WHITE PAPER

# XSOLLA IN-GAME STORE

v.1.0

# CONTENTS

# INTRODUCTION

## GLOSSARY

- **Players, end users, users**
  The target audience for Xsolla In-Game Store.
- **Partners, game developers, publishers**
  The game content providers.
- **Game, project**
  The content delivered to end users of Xsolla Subscriptions.
- **Xsolla In-Game Store, In-Game Store**
  The names of the product for which this document is intended.
- **Microtransaction**
  When you pay real world money for virtual items.
- **External ID**
  External unique ID of an entity

## WHAT IS XSOLLA?

Xsolla is the video game business engine with a set of tools and services that helps clients operate and sell more games globally. Serving only the video game industry, the Xsolla product suite caters to businesses from indie to enterprise, with: Xsolla Pay Station and its #1 Anti-fraud solution, Xsolla Partner Network, Xsolla Site Builder, Xsolla Store, Xsolla Login, and Xsolla Launcher. These tools work seamlessly to solve the complexities of distribution, marketing, and monetization so developers, publishers, and platform partners can increase their audience, sales, and revenue. Headquartered in Los Angeles, with offices worldwide, Xsolla operates as a merchant and seller of record for major gaming entities like Valve, Twitch, Ubisoft, Epic Games, and PUBG Corporation. For more information, visit www.xsolla.com.

# WHAT IS XSOLLA IN-GAME STORE?

Xsolla In-Game Store is a service for managing in-game purchases, often referred to as microtransactions. It functions as a marketplace within a game from which players can buy digital items for in-game use only. Xsolla In-Game Store is highly customizable and easy to implement via API or Unity/Unreal/Android SDK.

Key Features:

- sell virtual items for real or virtual currency:
    - in-game currency and currency packages
    - consumable items
    - nonconsumable items
    - nonrenewing subscription
    - bundles
- flexible price management
- integrated tools for marketing promotions
- works seamlessly with Xsolla Player Inventory, a player entitlement system that allows players to store and withdraw their premium rewards

# WHAT IS THIS DOCUMENT FOR?

Use this document for details about Xsolla In-Game Store technical implementation and check it for compatibility with your project. It describes the main Xsolla In-Game Store components, features, and their technical implementation.

In addition to this document, we recommend checking out the IGS eBook (part 1) and IGS eBook (part 2) - a free Xsolla guide that includes data-driven advice on the best way to add an in-game store to your game.

# PREREQUISITES

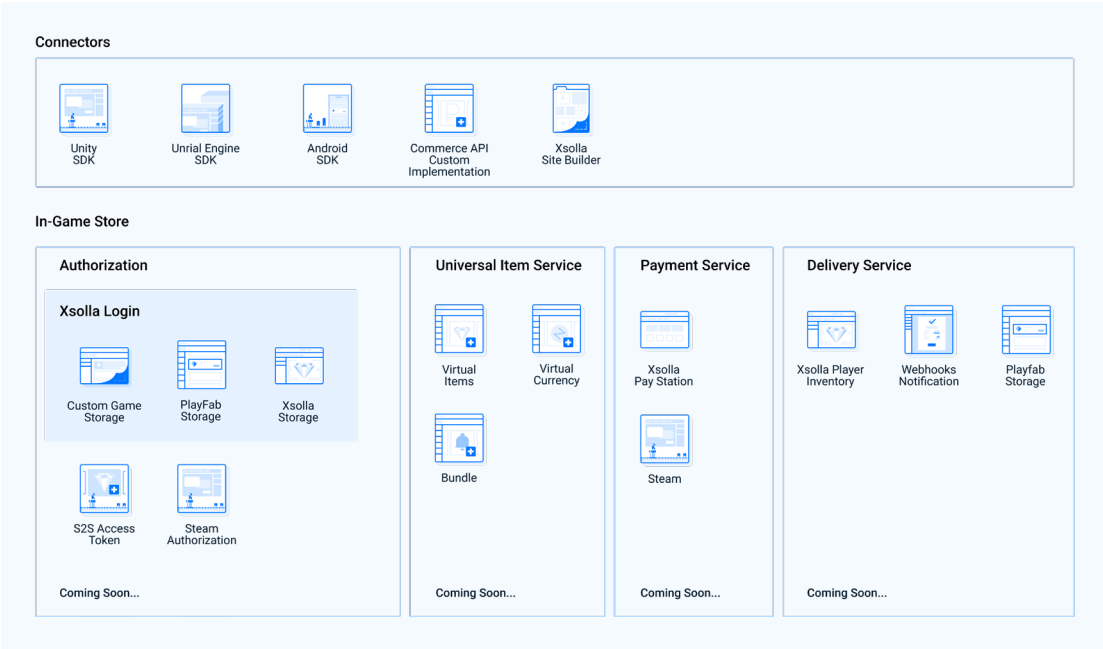
1. Create your Xsolla Publisher Account in order to configure Store.

Each project automatically generates its own ID when it gets created.
This **project_id** is used in API methods to identify a project instance that gets created in Xsolla Publisher Account.

# HOW IT WORKS

## ARCHITECTURE

Xsolla In-Game Store is a back-end solution that unites API calls, services, and other Xsolla products. Above all this is a set of connectors that allow you to implement Xsolla In-Game Store functions into your product and create the in-game store or a storefront.The user creates an account in the partner's service.




There are 4 functional blocks in Xsolla In-Game Store. Every block is responsible for the definite stage of end-user interaction with the store.

# User Authentication

A functional block is responsible for user authentication. Authentication allows you to identify the user and get all necessary information for working with their personal data, inventory, location, and also to get currency for catalogs, promotions, etc.

Possible authentication options:

- Authentication via Xsolla Login, if you don't have your own authentication system.
- You have 3 options for user data storage:
    1. Store data on the Xsolla side
    2. Use PlayFab data storage
    3. Store data on your side in a custom storage
- Authentication via Pay Station Access Token, if you have your own authentication system.


- Seamless authentication via Steam allows users to enter the game via the installed Steam application. You can opt for this option if you use one of Xsolla SDKs as a connector.

# Universal Item Service

A functional block that includes Commerce API and a universal item service. This block is responsible for storing data of the in-game store.

You can sell the following items via Xsolla In-Game Store:

- Virtual items
- Virtual currency
- Bundles

All entities that are used in the store logic have the same set of key fields and basic business logic, which simplifies the process of working with them. Every item type implements its own custom part of business logic.

## Payment Service

A functional block is responsible for payment of items. It gets user information, contents of the purchase, including the total sum, and prices of every item in the order. It also returns the order status.

Users can pay via:

- Xsolla Pay Station
- Steam, if the game is published there

## Delivery Service

This functional block works as a service for delivering items to users in different ways.

After successful payment, every item in the order is delivered to the user in the way that is specified in the project settings.

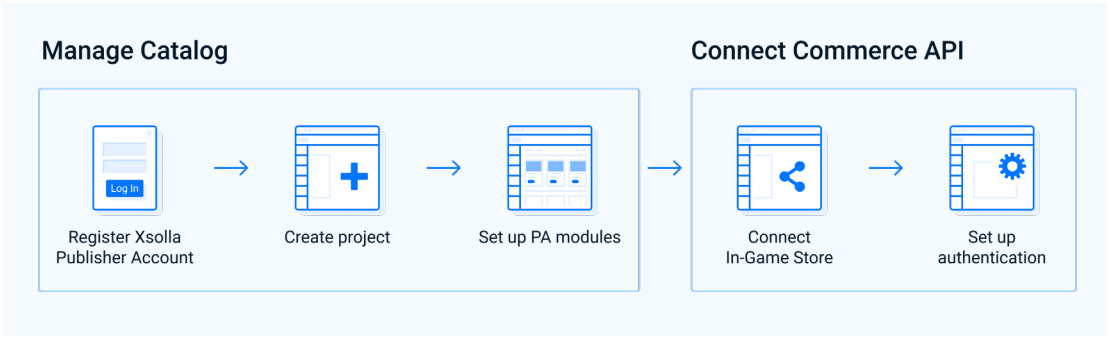
Available delivery methods:

- directly to Xsolla Inventory (default method)
- to PlayFab if you have configured the integration
- if the in-game store logic is implemented on your servers, you can configure the purchase webhooks and, after you process them, grant items to the inventory on your own.

## Connectors

Connectors are Xsolla services and products that allow you to implement Xsolla In-Game Store functions into your product and create the in-game store or the storefront.

# XSOLLA IN-GAME STORE INTEGRATION




Integration of Xsolla In-Game Store includes 2 necessary steps:

1. Manage your catalog.
2. Integrate Commerce API in your game using one of the suggested methods.

## 1. Manage your catalog

1. Register an Xsolla Publisher Account.
2. Create a project and configure it in Publisher Account.
3. Set up modules for **Virtual Items** and **Virtual Currency** in your Publisher Account.

## 2. Integrate Commerce API

You can integrate Xsolla In-Game Store in one of the following ways:

- in your game
  - via SDK
  - via Commerce API
- Create a web storefront
  - via Xsolla Site Builder; using Xsolla Site Builder
  - using the source code on GitHub

Integrating the game means creating an in-game store in the game UI. However, the web version of the store allows you to sell in-game items via websites and landing pages. The options aren't mutually exclusive and you can use both of them if you want.

## Integrating the game via SDK

You can use Xsolla SDK to integrate Xsolla In-Game Store into your game.

Xsolla SDKs are ready-to-use libraries for applications and games that allow game-developers to easily embed Xsolla services. The SDK doesn't impose restrictions on the visual style or design of interfaces for embedded services.

Xsolla provides SDKs for development on the following platforms:

- Unity
- Unreal Engine 4
- Android

SDKs come with multiple ready-to-go scenes that you can use as-is. All you need to do is change the UI only and set up a catalog of products in the store.

## Create web storefront via Xsolla Site Builder

Integration of In-Game Store allows you to create and sell in-game items on websites for games. You don't have to create a website via Xsolla Site Builder to connect In-Game Store to it.

You can sell the following types of products on landing pages:

- Virtual items
- Virtual currency
- Bundles

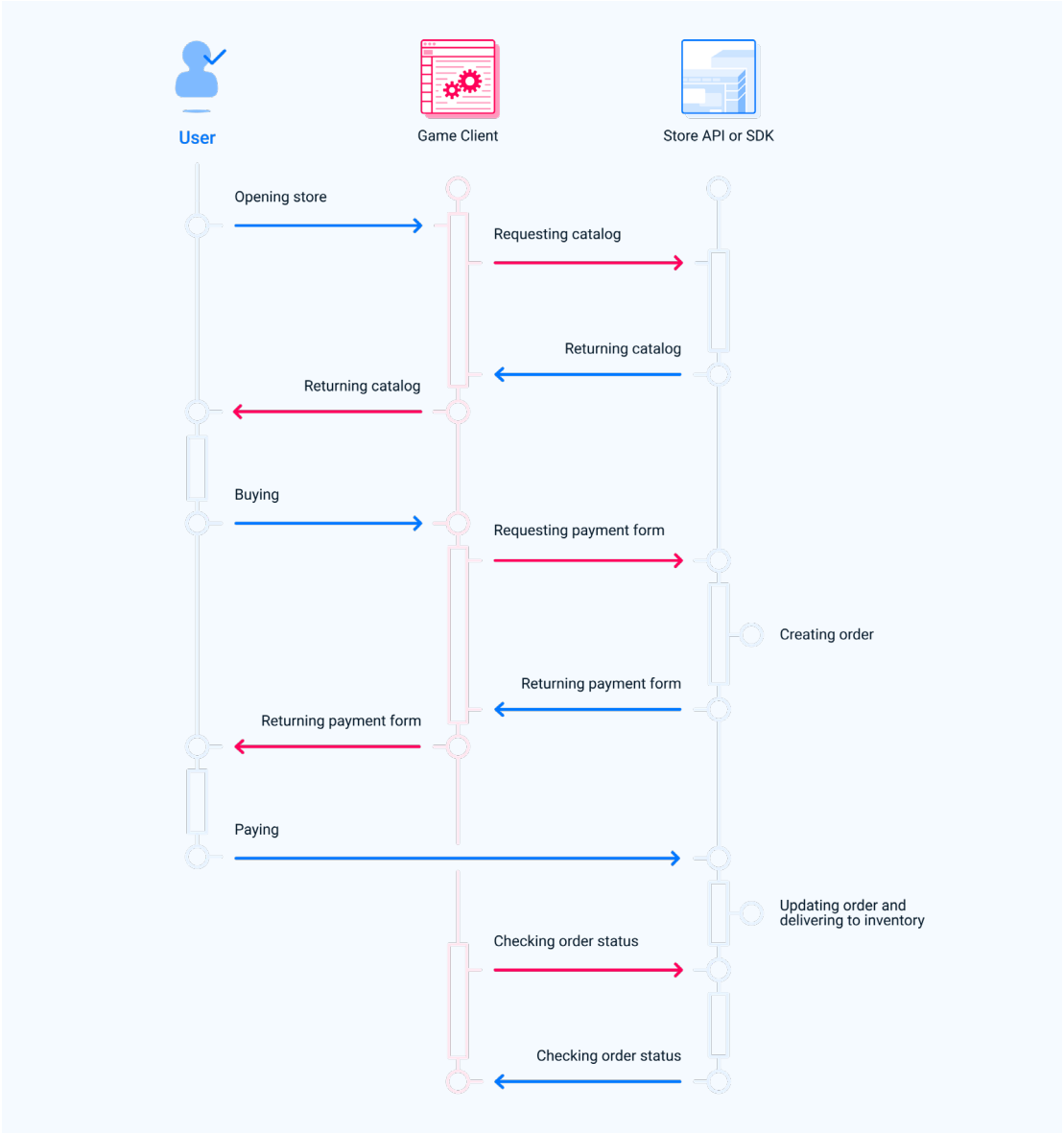
Use the Connecting a store to your site recipe for integration.

## Create web storefront using the source code on GitHub

You can connect Commerce API to a web store.

Xsolla provides access to store-demo source code on GitHub. You can use it as a basis or an example of the In-Game Store integration into a web store.

# ITEM PURCHASE FLOW IN XSOLLA IN-GAME STORE



# XSOLLA IN-GAME STORE FEATURES

## TYPES OF ITEMS

### General information

Types of items are sold in the store catalog.

Every item has:

- **Show in Store** toggle. It allows you to hide or show items in the catalog.
- Unique **SKU** that allows you to get the item via client API calls.

You can also set up attributes for items. Attributes are characteristics that give gamers additional information about an item that's listed in a store.

### Virtual Items

This feature allows you to sell in-game content for real or virtual currencies.

Key features:

- Set prices in real and virtual currencies.
- Set up catalogs with one or several levels.
- Configure brief and detailed item cards.
- Auto-detect user currency and country.

## Types of virtual items

### 1. Consumable virtual items

A consumable item is an item in the inventory that can be accrued or purchased repeatedly. It decreases in number once it's used.

Main features:

- You can replenish the stock of items.
- Users can have many units of the same item in the inventory.

**FOR EXAMPLE:**

grenades and bullets to attack the opponents, etc.

### 2. Nonconsumable virtual items

A nonconsumable item is an item in the inventory that can be accrued or purchased only once.

Main features:

- Users can have only one unit of this item in the inventory.

Restrictions:

- Can't be removed from the inventory by consuming on the client side. You can withdraw it only via a server method

**FOR EXAMPLE:**

access to a location, status, cosmetics, pre-installed DLC, NO ADS option for mobile games, etc.

### 3. Nonrenewing subscriptions

A nonrenewing subscription is a premium bonus that is a one-time purchase. The user needs to buy it again when it expires.

**FOR EXAMPLE:**

Battle Pass, Season Pass, temporary access to an in-game cosmetic item, item, or additional content.

Main features:

- Sell a subscription for an unlimited number of times.
- If the subscription is active and the user purchases it again, the newly purchased subscription isn't added to the inventory but its expiration date is extended.
- You can configure the subscription expiration time in Publisher Account.

Restrictions:

- A nonrenewing subscription is a nonconsumable item. It can't be removed from the inventory by consuming it on the client side. You can withdraw it only via a server method.

See the nonrenewing subscription recipe for details.

## Prices of virtual items

You can set up prices in real and virtual currencies for every item.

Key points:

- You can set up prices both in real and virtual currency for one item.
- You can set prices in several real and virtual currencies. Make sure you chose one default currency for the group of prices.

Restrictions:

- You can't create an item without a price.

Contact Xsolla managers if you want to create an item with zero value in price.

## Groups of virtual items

Groups allow you to set up catalogs with one or several levels.

Key features:

- The Get items list by specified group call is used to get the list of items of the definite group by its **external_id**.
- Create groups inside bigger groups.
- Items without a group are added to the **Ungrouped** group.

Restrictions:

- You can populate groups with virtual items only.

### Set up virtual items

To set up virtual items, you can:

- use functionality of Publisher Account
- use admin API calls

  To set up virtual items via API, you should use Basic authentication. A request to API must contain **Authorization:Basic <your_authorization_basic_key>** header, where **<your_authorization_basic_key>** is the **project_id:api_key** pair encoded according to the Base64 standard. You can get **project_id** and **api_key** values in Xsolla Publisher Account.

> API calls from the **Admin** subsection aren't created for building a catalog in the store on the client side. You shouldn't use them on landing pages, in web stores, and for in-game logic.

### Get catalog of virtual items on the client side

To get the catalog of virtual items on the client side, use the API calls from the **Catalog** subsection. They don't require authorization.

Client API calls allow you to get:

- a full list of items from all groups
- a list of items of one group by its **external_id**
- a list of groups in a project

## Virtual currency

This feature allows game developers to sell in-game currencies.

Key features:

- Sell packages of in-game currency.
- Manage in-game currency user balances.
- Auto-detect user currency and country.

## Hard virtual currency

Hard virtual currency is a virtual currency that users can buy only for real currency. Hard virtual currency is linked to a platform of purchase. To store currency separately on different platforms, currency for storage should be hard and you should pass the platform of purchase to inventory management API calls.

Restrictions:

- You can give currency a status of **hard** only during the first setup.
- You can sell packages of virtual currency with hard virtual currency only for real money.

## Virtual currency packages

This feature allows you to sell in-game currencies in packages, for which the price can be lower than if the currency was bought with a custom amount. For a package, you can define a special price that doesn't equal the amount of virtual currency in it. For example: 1 virtual coin = 1 USD. 500 virtual coins = 450 USD.

Virtual currency packages can have prices in both real and virtual currencies.

Restrictions:

- You can sell packages of virtual currency with hard virtual currency only for real money.
- You can add only one type of virtual currency to a package.

## Prices for virtual currencies and virtual currency packages

Key points:

- You should specify the prices for virtual currency and virtual currency packages at least in one real or virtual currency.

Restrictions:

- Contact Xsolla managers if you want to create a virtual currency and virtual currency package with zero value in price
- You can sell packages of virtual currency with hard virtual currency only for real money.

### Set up virtual currency and virtual currency package

To set up virtual currency and a virtual currency package, you can:

- use functionality of Publisher Account
- use admin API calls

  To set up virtual currency and packages of virtual currency via API, you should use Basic authentication. The request to API must contain **Authorization:Basic <your_authorization_basic_key>** header, where **<your_authorization_basic_key>** is the **project_id:api_key** pair encoded according to the Base64 standard. You can get **project_id** and **api_key** values in Xsolla Publisher Account.

  > API calls from the **Admin** subsection aren't created for building a catalog in the store on the client side. You shouldn't use them on landing pages, in web stores, and for in-game logic.

### Get catalog of virtual currencies and virtual currency packages on the client side

To get the catalog of virtual currencies and virtual currency packages on the client side, use the API calls from the **Catalog** subsection.

Client calls allow you to get:

- the list of virtual currencies in the project
- the list of virtual currency packages in the project

## Bundles

This feature allows you to sell bundles that include several items sold as a single unit.

Key features:

- Add items of different types to a bundle:
  - virtual currency (including the platform-dependent currency)
  - package of virtual currency
  - game keys for preselected DRMs
  - virtual items including nonrenewing subscriptions
  - bundles
- Configure the bundle pricing by a specified value in real and virtual currencies.

## Unpack bundle

A bundle automatically unpacks after:

- successful purchase
- successful granting to the player inventory via inventory management methods.

If a bundle includes another bundle, the included bundle automatically separates into standalone items.

## Return bundle

When the buyer returns the bundle by cancelling a transaction (for example), all granted items from the bundle are automatically withdrawn from the user. If they spent some of the items, the remaining items are still withdrawn.

## Set up bundles

To set up bundles, you can:

- use functionality of Publisher Account
- use admin API calls

  To set up virtual currency and packages of virtual currency via API, you should use Basic authentication.The request to API must contain **Authorization:Basic<your_authorization_basic_key>** header, where **<your_authorization_basic_key>** is the **project_id:api_key** pair encoded according to the Base64 standard. You can get **project_id** and **api_key** values in Xsolla Publisher Account.

  > API calls from the **Admin** subsection aren't created for building a catalog in the store on the client side. You shouldn't use them on landing pages, in web stores, and for in-game logic.

## Get catalog of bundles on the client side

To get the catalog of bundles on the client side, use the API calls from the **Catalog** subsection. Authorization isn't required.

Client calls allow you to get:

- a full list of bundles in the project
- a specified bundle by its **SKU**

  > See the bundles recipe for details.

# PROMOTIONS

This module includes a list of tools for setting up promotions for your products. It helps to get new players and increase sales.

Xsolla In-Game Store provides functionality for 4 types of promotions:

- Coupon promotions
- Promo codes
- Discount promotions
- Bonus promotions

## Coupons

Coupon promotions are a marketing tool for getting new clients and increasing sales. A player who uses a coupon gets a reward linked to this coupon (e.g., a virtual currency package, game key, or virtual item).

Players enter the coupon code on the website, directly in the game, or in the mobile application.

You can set the following virtual goods as a reward:

- package of virtual currency
- game key
- virtual item (including nonrenewing subscriptions)
- bundle

Main features:

- set up the number of redemptions:
  - general number for a campaign
  - number of redemptions for one user
- generate codes for coupons multiple times
- application of regional restrictions for promotions and goods linked to a coupon
- use keys for definite DRMs as a reward, or let users choose a DRM during a coupon redemption

Restrictions:

- Users can get a reward for a coupon with a nonconsumable item only once. More attempts to redeem a coupon will be declined, as there can be only one instance of the same nonconsumable item in the user inventory.
- If you linked a game key to a coupon and there are extra keys in a campaign that don't get sold, one of them will be emailed to a player who has a coupon. If there are no keys left in the campaign, the player's request to use a coupon will be declined.
- If the whole coupon campaign has regional restrictions, the user from the restricted region can't redeem a coupon.
- Regional restrictions are set on the Xsolla side upon request.
- If the items linked to a  coupon have regional restrictions, the user from the restricted region can't redeem a coupon and get these items.
- You can set a bundle as a reward only via the API.

## Coupon codes

A campaign may include one or several coupons. The coupon format has 2 options:

- Manually entered string (maximum is 128 characters).

**EXAMPLE:**

WINTER2021

The code is case sensitive. For example, winter2021 and WINTER2021 are different coupons.

- Generated string – string of numerals and Latin symbols (10 characters).

**EXAMPLE:**

7d9ekfl5hj

Coupon codes are generated after creating a coupon campaign during the editing of the promotion in Publisher Account, or via Generate coupon codes and Create coupon code API calls. You can generate codes multiple times. Massive uploading of codes is possible on the Xsolla side upon request.

You can get the list of codes via the Get coupon codes API call.

## User flow

1. Users enter the coupon code on the website, directly in the game, or in the mobile application.

2. The item that is linked to a coupon is granted to the player in one of the following ways:

    - directly to the inventory (a virtual item or virtual currency package)
    - by email
    - by game entitlement system (a game key)

## Integration flow

1. Set up a promotion campaign in one of the following ways:

    - use functionality of Publisher Account
    - use API calls

    To set up a promotion campaign via API, you should use Basic authentication. Request to API must contain **Authorization:Basic <your_authorization_basic_key>** header, where **<your_authorization_basic_key>** is the **project_id:api_key** pair encoded according to the Base64 standard. You can get **project_id** and **api_key** values in Xsolla Publisher Account.

2. Implement the Get coupon rewards API call to get the list of items that the user receives after redeeming a coupon. This action doesn't redeem a coupon. This step allows you to display the list of items that the user will receive on the client side. Depending on the is_selectable parameter value in the response, there are 2 possible flows:

    - If **is_selectable=true** in the response, the reward includes game keys with DRMs and the user should choose one of them. The client side should have a DRM choice feature. After the user picks a DRM, the result is passed to the redemption method.
    - If **is_selectable=false**, you shouldn't pass additional parameters to the redemption method, the redemption method is called instantly.

3. Implement the Redeem coupon code API call to use a coupon. If the user chooses a DRM key on the previous step, you should pass an object with the choice results in the request body.

    The system checks the campaign's expiration time and limitations, whether there are available keys and regional restrictions, and, if the redemption is possible, delivers the reward to the end-user.

See the Coupons recipe for details on coupon promotions and their configuration.

# Promo codes

Promo codes are a marketing tool for getting new clients and increasing sales. During this promotion, the user gets a promo code that they can use in the cart UI.

After redeeming a promo code, the user can get:

- a percentage discount on items in the cart
- a bonus item:
    - virtual currency package
    - game key
    - virtual item (including nonrenewing subscriptions)
    - bundles

Main features:

- simultaneous provision of a discount and a bonus by one promo code
- set up the number of redemptions:
    - general number for a campaign
    - number of redemptions for one user
- After the user applies a promo code, they can change the cart's content. The discount automatically applies to the added items and recalculates the cart's total.

Restrictions:

- Users can activate a promo code in the cart UI only, the promo code is used only after the payment.
- Users can apply only one promo code to one cart and only if there are items in the cart.
- If the whole promo codes campaign has regional restrictions, the user from the restricted region can't redeem a promo code.
- Regional restrictions are set on the Xsolla side upon request.
- If a promo code includes an item with regional sale restrictions, the user from a restricted region can't get the items linked to this promo code.

## Promo codes

A campaign may include one or several promo codes. The promo code format has 2 options:

- Manually entered string (maximum is 128 characters).

**EXAMPLE:**

WINTER2021

The code is case sensitive. For example, winter2021 and WINTER2021 are different promo codes.

- Generated string – string of numerals and Latin symbols (10 characters).

**EXAMPLE:**

7d9ekfl5h4

You can get the list of codes via the Get codes of promo code API call or via Publisher Account.

## User flow

1. The user adds items to the cart.
2. The user enters the promo code in the corresponding field in the cart UI.
3. The cart's total is recalculated.
4. The user pays for the cart and redeems a promo code by it.
5. If the promo code included some bonus items, they are delivered to the user in one of the following ways:
    1. directly to the inventory (virtual item or a virtual currency package)
    2. via email (a game key)
    3. directly to the entitlement system (a game key)

## Integration flow

1. Set up a promotion campaign with promo codes in one of the following ways:

   - use functionality of Publisher Account
   - use API calls

   To set up a promotion campaign via API, you should use Basic authentication. The request to API must contain **Authorization:Basic <your_authorization_basic_key>** header, where **<your_authorization_basic_key>** is the **project_id:api_key** pair encoded according to the Base64 standard. You can get **project_id** and **api_key** values in Xsolla Publisher Account.

2. Implement the Get promo code rewards API call to get the list of items that the user receives after redeeming a promo code. This action won't redeem a promo code.

   This step allows you to display the list of items that the user will receive on the client side. Depending on the **is_selectable** parameter value in the response, there are 2 possible flows:

   - If i**s_selectable=true** in the response, the reward includes game keys with DRMs and the user should choose one of them. The client side should have a DRM choice feature. After the user picks a DRM, the result is passed to the redemption method.
   - If **is_selectable=false**, you shouldn't pass additional parameters to the redemption method, the redemption method is called instantly.

3. Implement the Redeem promo code API call to use a promo code. If the user chooses a DRM key on the previous step, you should pass an object with the choice results in the request body.

   The system checks the campaign's expiration time and limitations, whether there are available keys and regional restrictions, and, if the redemption is possible, changes the cart according to promotion conditions.

   See the Promo codes recipe for details on promo codes promotions and their configuration.

# Discounts

Discount promotions are a marketing tool that allows you to set up a discount on an item or group of items.

Main features:

- Apply an unlimited number of discount promotions to one item.
- Create a catalog of items with automatic calculation of pricing for them. Methods for building a catalog return the price object for every item. This object contains prices for items with a current discount and without them.


```
«price»: {
«amount»: «2.9900000000000000»,
«amount_without_discount»: «2.9900000000000000»,
«currency»: «USD
}
```


Restrictions:

- You can set the discount only in percentage value in the range from 1 to 99.
- If the item has several discount promotions, they are not added together but are applied one by one.
- You can't apply regional restrictions to discount promotions.
- The promotion is created with the "is_enabled":false status. After you create it, activate the promotion manually from Publisher Account or via API.

## Integration flow

Set up a promotion campaign with discounts in one of the following ways:

- use functionality of Publisher Account;

If you set up discount promotions via Publisher Account, you have the following features:

- Set up a discount with the integer percentage value.
- Set up discount promotions that are available for a limited time period.
- Set up discounts that apply to one or several items of one of these types:
  - virtual currency packages
  - virtual items (including nonrenewing subscriptions)
  - game keys

The following is a list of actions that aren't available in Publisher Account:

- Set up discounts for physical goods.
- Set up discounts with fraction values.
- Set up discounts with an unlimited time period.
- Set up discounts for different item types in one discount promotion.
- Edit discount promotions with the Active status.

- via API calls

To set up a promotion campaign via API, you should use Basic authentication. The request to API must contain **Authorization:Basic<your_authorization_basic_key>** header, where **<your_authorization_basic_key>** is the **project_id:api_key** pair encoded according to the Base64 standard. You can get **project_id** and **api_key** values in Xsolla Publisher Account.

Extended features, if you set up a campaign via API:

- Edit discount promotions with the Active status
- Set up a discount on all items in the project
- Set up a discount with fraction values
- Set up a discount on items of different types
- Set up a discount on definite DRMs for a game
- Set up a discount on physical goods
- Set up a discount promotion without specified time period

## Customize conditions of discount promotion

You can set up a discount that is applied only if some conditions are met. For example:

- a discount that is applied to a definite total sum of a cart
- a discount that is applied only if the cart has enough items
- a discount on one item if users buy another item

To set up discount promotions with conditions, contact your Account Manager.

See the Discounts recipe for details on discount promotions and their configuration.

# Bonus promotions

Bonus promotions are a marketing tool that allows you to set conditions for giving bonus items to users.

Main features:

- Grant bonus for purchasing a definite item

Restrictions:

- You can't apply regional restrictions to bonus promotions.
- You can configure bonus promotions only via API calls.

## Integration flow

You can configure bonus promotions only via API calls.

To set up a promotion campaign via API, you should use Basic authentication. The request to API must contain **Authorization:Basic <your_authorization_basic_key>** header, where **<your_authorization_basic_key>** is the **project_id:api_key** pair encoded according to the Base64 standard. You can get **project_id** and **api_key** values in Xsolla Publisher Account.

## Customize conditions of a promotion with bonuses

You can set up custom conditions for a promotion with bonuses. For example:

- Bonus for purchasing an item with a definite attribute (Items in the X group, or items).
- Bonus for a definite number of items in the cart. (Purchasing 4–6 units of one item gives 10% discount, buying more than 6 units gives users a 20% discount).

To set up custom conditions, contact your Account Manager.

# STORE CATALOG MANAGEMENT

## Pricing and regional restrictions

Catalog pricing

Key points:

- The catalog has one currency, i.e., if you request a catalog from the server, the price of all items will be returned in one currency.
- The catalog is built in real currency.

Restrictions:

- All items must have a default price.

You should define a default real currency to show prices of items in it. (e.g., USD). It helps to avoid errors when building a catalog.

### Price in real currency

```
«prices»: [{
   «amount»: string,
   «currency»: string,
   «is_default»: boolean,
   «is_enabled»: boolean
 },
 ...
]
```

Every price in real currency is an object with the following specified values:

- **currency** – currency in the three-digit format per ISO 4217
- **amount** – item's price in the currency from the **currency** field
- **is_default** – default price that is used to build a catalog if no price in the user's currency is specified
- **is_enabled** – the price is enabled and is used for calculating the catalog currency

Any item can have only one default currency.

You can configure prices either via Publisher Account or API calls.

## Prices in virtual currency

```
«vc_prices»: [{
   «sku»: string,
   «amount»: integer,
   «is_default»: boolean,
   «is_enabled»: boolean
 },
 ...
]
```

Every price in virtual currency is an object with the following specified values:

- **sku** – SKU of a virtual currency that is used for defining the price
- **amount** – price of the item in the currency from **sku**
- **is_default**
- **is_enabled**

Any item can have only one default virtual currency.

You can configure prices either via Publisher Account or API calls.

## Regional restrictions and prices

**Region** is a set of countries.

**EXAMPLES OF REGIONS:**

Worldwide: all countries excluding Mainland China, Hong Kong, Macao, and Taiwan.

Europe: all countries in Europe.

**Regional pricing** is a mechanism that allows you to set different prices for the same items depending on the buyer's country/region.

**EXAMPLE:**

The game costs 59EUR in the UK, but in Poland it costs 39EUR.

**Regional lock** is a mechanism that limits the following actions in the restricted region:

- buying the item
- getting the item
- promotion activation

Key points:

- Regions for local prices and regional locks are the same.
- Items without a region are automatically given the Worldwide region and have no restrictions.
- User's region is defined by the region of their account. If the account doesn't have information about the region, the region should be passed in a token. If there's still no region, we suggest the region by user IP.

Restrictions:

- One region can have several prices, but only one can be a default price.
- You can't configure regions, regional locks, and regional prices in Publisher Account.
- You can configure regions and regional prices for virtual items and virtual currency only via API calls.

**Set up regional prices and restrictions**

- Regions are configured by Xsolla managers upon request.
- Via API, you can configure only regional locks and prices for bundles.
- Regional locks and prices for coupon promotions and promotions with promo codes are configured by Xsolla managers upon request.
- If the item has a region, but doesn't have a regional price, the item will be unavailable in this region.

Configuration flow

1. Send to Xsolla a request for a regional locks and regional prices configuration. In this request specify:
    1. list of regions and countries in them
    2. for items, a list of currencies and prices for regions and countries with the default price
    3. if you need regional locks for coupon promotions or promo codes promotions, add the list of restrictions for regions
2. Xsolla creates regions, countries for them, regional prices, and regional restrictions.

### Select real currency for catalog

1. Filter the items that have price only in virtual currency.

2. The currency is selected according to a priority:

- **Currency passed in the GET-request parameter**
  All items in a catalog should have prices in this currency, otherwise the lower priority currency is used.

- **Currency of user's region**
  All items in a catalog should have prices in this currency, otherwise the lower priority currency is used.

- **Default item price**
  The default currency of the first item in the list is used. Items without prices in this currency aren't displayed.

Catalog has one currency. Specify the same default currency for all items in a catalog.

## Groups

Groups allow you to set up catalogs with one or several levels.

Currently we support group distribution only for virtual items. Read more in Groups of virtual items.

## Attributes of items

Attributes are characteristics of items that give gamers additional information about an item that's listed in a store. One item can have several values of one attribute. Examples of attributes: genre, item's color, and item's size.

Main features:

- specify additional characteristics of an item

- link items to events and conditions in a game's logic

Limitations:

- You can't set up an attribute for an item without specifying a value for this attribute.

- You can't specify more than 6 values of the same attribute for one item. To change this limit, contact your Account Manager.

- You can't set up more than 20 attributes for one item. To change this limit, contact your Account Manager.

### Examples of attributes

| ITEM TYPE | ATTRIBUTE NAME | ATTRIBUTE VALUE |
|---|---|---|
| GAME KEYS | Game type | New / Early access / Coming soon / Specials |
| | Genre | Action / Adventure / Casual / Simulation / Strategy / RPG / Sports / MMO / Racing / Puzzle / Horror / Sci-Fi / Shooter / Retro / Arcade / Platformer / Survival / Visual Novel / Turn-Based |
| | Platform | Windows / MacOS / Nintendo Switch / Android / iOS / PS4 / PS5 / Xbox One / Xbox Series X / Steam |
| VIRTUAL ITEMS | Quality | Epic, Legendary, Common |
| | Color | Red / Blue / Green |
| | Event | <New year> |

## Set up attributes

Attributes are created for a project. Every attribute has the following parameters:

- External ID
- List of values
- Name (specified for every language used in a project)

> The number of attributes in a project is unlimited.

To create and set up attributes, you can:

- Use API calls:
  - Create attribute
  - Update attribute
  - Get specified attribute
  - Delete attribute
- Get the list of attributes for administration

## Configure attribute values

Every attribute value has the following parameters:

- External ID of a parent attribute
- External ID of a value
- Name (specified for every language used in the project)

To create and set up attribute values, you can:

- Use the functionality of Publisher Account: **Store > Catalog Management > Attributes**
- Use API calls:
  - Create attribute value
  - Delete all values of attribute
  - Update attribute value
  - Delete attribute value

When you receive an item via API calls, you will get the list of attributes and their values connected to the item in the item parameters. You don't need to use another API call for getting information about attributes.

If you delete a value, all its connections with items will be lost. You can't restore this data. A value will be removed from all ites it's connected to.

# EXTENSIONS

## PlayFab

This feature allows you to set up the integration between Xsolla In-Game Store and PlayFab. PlayFab is a server-based platform for game developers that allows you to implement your game economy without coding.

You can integrate PlayFab to provide a server side for your project and receive notifications after successful purchase of items via Commerce API. The purchased item is granted by passing the user PlayFab ID in the authorization token during the request to Commerce API.

Main features:

- Complex server-based solution for releasing your games.
- Support for many gaming platforms.
- Deep analytics for tracking user behaviour.

Restrictions:

- Only supports virtual items.
- Selling of Xsolla Virtual Currency isn't supported. It's implemented via selling bundles with virtual currencies on the PlayFab side.

**Integration flow**

1. If you don't have PlayFab integrated, do the following:
    1. Create your PlayFab account.
    2. Create the game.
    3. Get the unique game Title ID.
    4. Get PlayFab secret key.
2. 1. Send your Title ID and PlayFab secret key to your Account Manager.
    2. Specify PlayFab ID in the token in one of the following ways:
        a. By integrating Xsolla Login with PlayFab.
        b. By generating a token with the **user.playfab_id** parameter.

Contact your Account Manager to implement refund notifications.

# PLAYER INVENTORY

**Xsolla Player Inventory** allows you to:

- Use Xsolla API to grant and revoke premium items and currency in the entitlement system.
- Synchronize all user purchases and premium rewards on all platforms.

## Inventory management

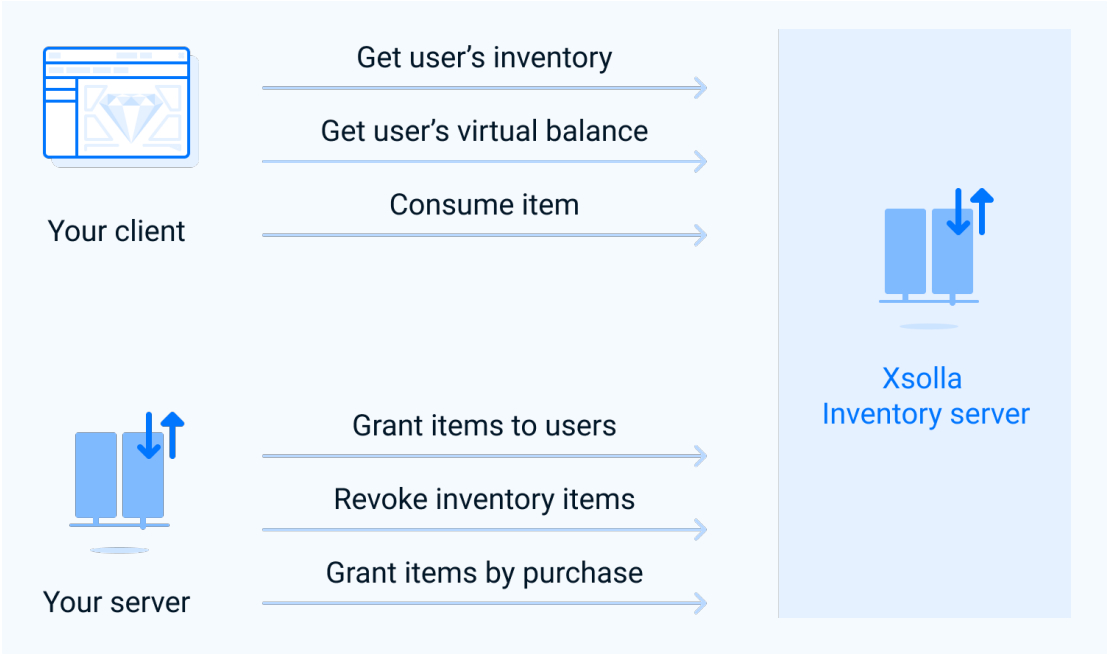
Player Inventory manages the user's inventory by the user ID. User identification is implemented through Xsolla Login. If you configured your own identification system, you can use the Pay Station Access Token for client API methods.

To work with Player Inventory API methods, authentication must be configured.

The user's inventory may include both the consumable and nonconsumable items.

- Consumable items can be purchased and added to the inventory several times and removed from it through the game client.
- Nonconsumable items can only be purchased and added to the inventory once. Cancelling the purchase and removing it from the inventory are carried out through the game server.

The scheme below illustrates the following principle of operating with Player Inventory:



1. **Getting the inventory**

    1. The user is identified in the game by their Xsolla Login account or your identification method.

    2. Your application (client) calls the Get user's inventory method to get a list of purchases and user rewards and the Get user's virtual balance method to get the balance of virtual currency.

    3. The Xsolla Inventory server returns information about purchases, rewards, or the balance currently owned by the user to your client.

2. **In-game purchase via Xsolla In-Game Store**

    1. The user is identified in the game by their Xsolla Login account or your identification method.

    2. The user makes a purchase via Xsolla In-Game Store.

    3. The content purchase event is sent to the Xsolla Inventory server, and then the purchased items are automatically granted to the user's inventory.

    > This algorithm is fully implemented on the Xsolla's side and does not require operations on the the game server's side.

3. **In-app purchase via a third-party publishing platform (Steam, PlayStation, etc.)**

    1. The user logs in to the game using the platform account.

    2. The user makes a purchase through the platform payment system.

    3. The game server calls the Grant items by purchase to users method to grant the purchase on the Xsolla Inventory server sending them a unique user ID within the game

    4. The Xsolla Inventory server adds a purchase to the user's inventory with the specified ID.

    > On the server side of the game, it is necessary to ensure the synchronization between user accounts on various publishing platforms.

4. **Granting rewards to the user**

    1. The user is identified in the game using their Xsolla Login account or your identification method.

    2. Your server calls the Grant items to users method to grant a reward, sending a unique user ID inside the game.

    3. The Xsolla Inventory server adds a reward to the user's inventory with the specified ID.

    > On the server side of the game, it is necessary to ensure the synchronization between user accounts on various publishing platforms.

5. **Inventory synchronization**

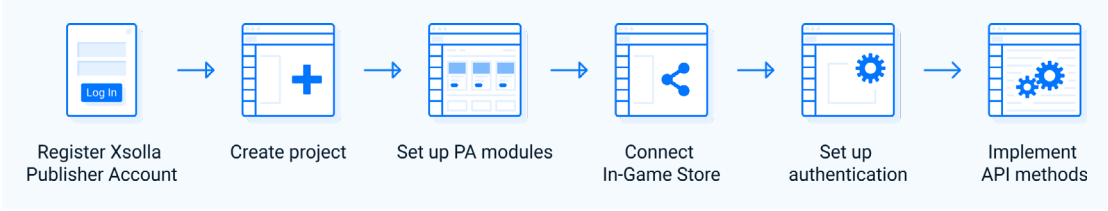
    1. The user is identified in the game by their Xsolla Login account, your identification method, or a third-party platform account.

    2. The game server synchronizes user accounts if necessary.

    3. The game client calls the Get user's inventory method to check which in-game purchases are available for use and sends a unique user ID.

    4. The Xsolla Inventory server returns a list of available items for the specified ID.

## Integration flow




1. Register an Xsolla Publisher Account.

2. Create and configure a project in Publisher Account.

3. Set up **Virtual Items** and **Virtual Currency** modules in your Publisher Account.

4. Connect In-Game Store via API or Unity/Unreal SDK.

5. Set up authentication.

6. Implement methods for inventory management.

> See the Xsolla Player Inventory recipe for details.

# Cross-platform inventory

Cross-platform Inventory allows game developers to synchronize a user inventory within a game on different platforms (Steam, EGS, PlayStation, XBox, etc.), manage user inventory and balance, and collect data about them.

The following steps ensure the synchronization:

1. The player links their platform accounts to the main account that is used to identify the player on different platforms.

2. Xsolla provides the mechanism for linking platform accounts to the main account and managing the cross-platform inventory synchronization.

The solution includes the following components:

- game client
- game server
- web application with an account linking UI
- **Xsolla Login** for providing the main account and a linking mechanism
- **Xsolla Player Inventory** for storing the user inventory

## User flow

To link a platform account (Steam or EGS) to the main account, you need to implement user identification in the game by using the Xsolla Login account or the algorithm for linking accounts on game console platforms.

The algorithm for linking the game console account to the main account is the following:

1. The player enters the game on the game console platform for the first time.

2. The game shows the message that offers to link the platform account to the main account.

3. If the player confirms the action:

    1. They are redirected to the side application (e.g., the game website or mobile app), where they can authenticate or create the main account.

    2. After successful authentication in the app, the player receives an alphanumeric code for linking the accounts.

4. The player returns to the console version of the game and enters the code.

5. The main and platform accounts link and the game console shows the confirmation message.

6. The player inventory on the platform synchronizes with the inventory linked to the main account.

The premium currency (currency purchased for real money) does not synchronize in the inventories. According to the console platforms' regulations, premium currency purchased via PlayStation, Xbox, and Nintendo Switch game consoles does not stack and is stored separately.

Premium currency is implemented via hard virtual currency functionality.

You can't unlink the accounts once they are linked. Also, if the user refuses to link the account when entering the game for the first time, they will not be able to link them in the future.

## Integration flow

To connect a cross-platform inventory:

1. Connect the Player Inventory. Implement basic access authentication for working with the server methods, and Xsolla Login authentication based on the OAuth 2.0 protocol for working with the client methods. OAuth 2.0 protocol based authentication is also required for accessing the Login API methods and implementing the account linking methods.

2. Implement the account linking UI including:

    1. game server-based authentication and inventory management

    2. game client-based authentication and inventory management

    3. account linking methods

If you have already implemented the account synchronization on the game server side, setting up the Player Inventory will be enough for the user inventory synchronization.

See the Cross-Platform Inventory recipe for details.

# RELATED XSOLLA PRODUCTS

## XSOLLA LOGIN

Xsolla Login is a single sign-on tool that authenticates and secures user passwords on behalf

of partners who develop video games. It creates a seamless, one-click user registration experience through 30+ third-party authentication providers that offer players convenient, safe, and fast methods for signing up or logging in to all of their favorite games.

Xsolla Launcher uses Xsolla Login API methods and Xsolla Login Widget for user registration

and password recovery. Xsolla Login API methods are also used for authentication. You can customize the authentication interface in the same way as other elements of Launcher Desktop. The data between Launcher and Login is transferred via Login API and Launcher API.

See Developer Documentation to get more information about Xsolla Login.

## XSOLLA BUY BUTTON

Xsolla Buy Button is a product related to Xsolla Ii-Game Store. Buy Button is a comprehensive solution for digital goods selling. Allows you to grant players access to your game when it's live or drive early adoption with pre-orders.

Supports the Promotions and Bundles feature.

See Developer Documentation to get more information about Xsolla Buy Button.

# XSOLLA SITE BUILDER

Xsolla Site Builder is an agile website construction tool that lets developers create their own online space to sell games and in-game goods directly to players, all without the need to invest

in costly web development resources. Developers can take full control of their online presence with mobile-responsive landing pages and advanced analytics that help them discover new players and deliver targeted outreach.

The Xsolla In-Game Store solution integrates with Xsolla Site Builder so partners can easily set up the storefront for selling in-game items on the game website.

See Developer Documentation to get more information about Xsolla Site Builder.

# XSOLLA PUBLISHER ACCOUNT

Xsolla Publisher Account provides the UI for setting up catalog for Xsolla In-Game Store.
The interface lets you manage the following settings:

1. Project settings
2. Store modules
    - Virtual Currencies
    - Virtual Items
    - Bundles
    - Promotions

# REFERENCES & LINKS

1. Publisher Account: https://publisher.xsolla.com/
2. Xsolla In-Game Store API References: https://developers.xsolla.com/store-api
3. Xsolla In-Game Store Developer Docs: https://developers.xsolla.com/doc/in-game-store/
4. Xsolla Login Developer Docs: https://developers.xsolla.com/doc/login/
5. Xsolla Buy Button Developer Docs: https://developers.xsolla.com/doc/buy-button/
6. Xsolla Site Builder Developer Docs: https://developers.xsolla.com/doc/site-builder/
7. Xsolla Store SDK Developer Docs: https://developers.xsolla.com/#sdk

# SUMMARY

Xsolla In-Game Store is a constantly developing product. We are constantly at work to make improvements and pay close attention to all feature requests. Integration with other Xsolla products will give you an opportunity to fully customize the solution. Learn more about other Xsolla products at xsolla.com or contact our integration team at integration@xsolla.com.

ENJOY THE GAME