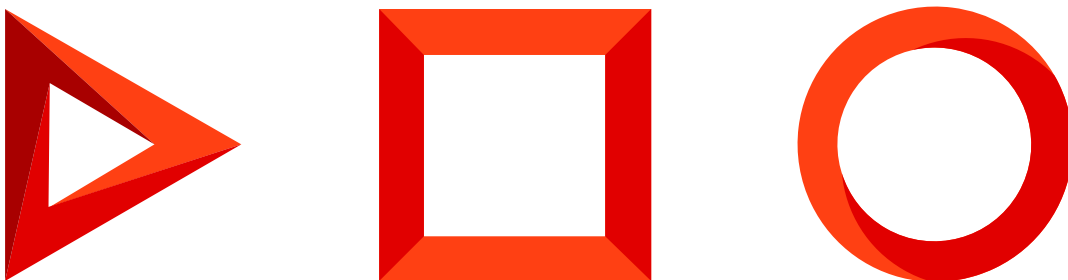


Marketplace app development

Requirements for Marketplace app

Version 8.0



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Table of Contents

Requirements for Marketplace app	4
Localization requirements	4
Functionality requirements	4
Front-end requirements	5
Security requirements	5
Compatibility requirements	6
Metadata requirements	6
Package requirements	7
Demo records requirements	7

Requirements for Marketplace app



Medium

Make sure the Marketplace app meets the following **groups of requirements**:

- localization requirements
- functionality requirements
- front-end requirements
- security requirements
- compatibility requirements
- metadata requirements
- package requirements
- demo records requirements

Localization requirements

- English version is mandatory.
- Translations to other languages are optional.
- Translate the following **elements** if users can use the Marketplace app in other languages:
 - app UI, for example, sections, details, fields, dashboards, notifications, etc.
 - localizable strings
 - bound data, for example, lookup values, demo data, system settings, etc.
- Translation of system data is optional. For example, headings and elements of business processes, headings of objects and schemas, and log data.
- The Marketplace app package must contain the localizations (Developer profile → [*General information*] tab → [*Localization*] property). Learn more in a separate article: [Manage UI languages](#).

Functionality requirements

- Use the Marketplace app to solve a business task that cannot solve with the basic development tools, or this solution is time-consuming or optimizes the business task solution.
- Do not duplicate base Creatio functionality in the Marketplace app. If the Marketplace app extends base functionality, use existing Creatio functionality.

Examples that comply with the requirement:

- Extend the [*CTI panel*] tab functionality of the base [communication panel](#) that use for the call management during the new telephony connector development.
- Extend the [*Chats*] tab functionality of the base communication panel used for the chat management

during the new chat connector development. You need not develop a UI that displays chats.



- Develop a custom contact action and add it to the [*Actions*] button menu of the contact page toolbar. Replacing the CSS styles and developing the new button in another place on the contact page are unnecessary.

Front-end requirements

- Requirements for the **icon** of the Marketplace app section are as follows:
 - Style: a flat white image without small elements or lines, background must be transparent.
 - Format: *.png or *.svg.
 - Size: 38x38 pixels or smaller. Note that the required icon size without border margins is 32x32 pixels.
 - We recommend selecting an icon of the Marketplace app section from the [section icon library](#).
- Capitalize the first word of the UI element name. Using the capital letter for each word of the interface element name is unnecessary.

View an example of a custom field name below.

Example of a custom field name

	
[<i>Serial number</i>]	[<i>Serial Number</i>]

- Use the basic page and navigation elements of the Creatio instance for the Marketplace app interface. For example, fields, buttons, tabs, etc.
- Enable the user to use the base Creatio functionality in its original form, whether implementing the new functionality or extending the basic Creatio functionality in the Marketplace app.
- Follow these **principles** when you develop the Marketplace app functionality:
 - **Clarity.** Develop a clear and logical interface to help users orient in the Marketplace app. Name UI elements as calls to action, for example, add, upload, etc.
 - **Consistency.** Design the app functionality so that it solves a specific business problem in its entirety as opposed to improving the experience of an individual functional or organizational role.
 - **Specificity.** Let the user use the additional technology implemented in the Marketplace app, for example, keyboard and mouse input, touch input, etc.

Security requirements

- The Marketplace app must not decrease Creatio performance.
- The Marketplace app must follow information security best practices. Use the recommended API Creatio calls in the Marketplace app. Ensure that Open Web Application Security Project (OWASP) Top 10 vulnerabilities are absent in the Marketplace app. Do not use components with known vulnerabilities. Learn more about vulnerabilities on the [official OWASP website](#).

- Use the law way to get data access, and do not send data from the Creatio instance in the Marketplace app. Describe the data sending on the Marketplace app page and restrict the data sending before the user authorization when you integrate with external apps.
- Do not use the anonymous web services without an alternative authorization type in the Marketplace app. Learn more in a separate article: [Custom web services](#).
- Do not use external libraries from unknown sources in the Marketplace app.
- Use package install scripts to execute CRUD operations with data as part of the package installation process instead of SQL scripts. Install scripts execute operations with data using the `Entity` class. Learn more in a separate article: [Customize delivery process](#).

Compatibility requirements

- Implement compatibility with the MS SQL and PostgreSQL databases for new Marketplace apps (the Developer profile → the [*Installation and setup*] tab → the [*Compatibility*] block → the [*DBMS compatibility*] property). Implementing compatibility with the Oracle database for Marketplace apps is optional.
- The Marketplace app must support .NET Framework (Developer profile → [*Installation and setup*] tab → [*Compatibility*] block → [*Platform*] property). .NET Core support is optional.
- Implement compatibility with the latest Creatio version of the Marketplace app.
- Implement compatibility with the existing Marketplace app updates with the previous app version.
- Describe all implemented corrections and modifications when you release the new version of the existing Marketplace app (the Developer profile → the [*Packages and updates*] tab → the [*Whats new?*] property).
- The Marketplace app must be compatible with base Creatio products (Developer profile → [*Installation and setup*] tab → [*Compatibility*] block → [*Product compatibility*] property) and versions (Developer profile → [*Installation and setup*] tab → [*Compatibility*] block → [*Version compatibility*] property). If you use the Studio Creatio platform as a base product, the Marketplace app is compatible with all base Creatio products.
- Support all versions of the Marketplace that the users use with actual Marketplace app licenses or the paid support package.
- Use the file management API in your Marketplace app instead of direct SQL queries. The API lets you access files stored in the database and external storage. Learn more about the API in a separate article: [API for file management](#).

Metadata requirements

- Do not use the base metadata template generated automatically.
- Make sure the [*Please name your application*] field value entered when creating the app matches the app name on Creatio Marketplace.
- Select any icon in the [*Select icon and color*] block or use custom icon. Make sure the icon matches Marketplace app functionality.
- Select any icon color in the [*Select icon and color*] block.
- If you use custom icon for the Marketplace app, make sure the icon meets the following requirements:
 - Style: a flat white image, background must be transparent.

- Size: 32x32 pixels.
- Format: *.svg.

Package requirements

- Release the Marketplace app as a single Creatio app and/or setting in an external app.
- Fill out the Marketplace app metadata before delivery. Learn more in separate articles: [Set up the app metadata](#), [Add metadata to a published Marketplace app](#).
- Convert simple packages and project packages that contain Marketplace app functionality to assembly packages. To do this, follow the instructions in a separate article: [Package conversion](#).
- Deliver the implemented customization as a package. Do not customize Creatio beyond the configuration, for example, create or change files in the Creatio file system.
- The Marketplace app functionality must include the unique developer ID (prefix) (Developer profile → [*General info*] tab → [*Marketplace developer settings*] block → [*Developer prefix*] property). Do not develop the Marketplace app without a prefix or with the `Usr` prefix.
- Upload the Marketplace app as a *.zip archive (Developer profile → [*Packages and updates*] tab → [*Add file*] property) that contains one or more *.gz packages.
- Add all *.gz package archives in the root folder of the *.zip archive. Do not use subfolders.
- Name the *.zip archive using the following template: `Marketplace-app-name-Package.Version.Number`.

View an example that names the *.zip archive of the custom Marketplace app below.

Example of a custom *.zip archive

	
<code>Salesloft-connector-1.15.zip</code>	<code>Salesloft_connector_01.06.2022_10_16.zip</code>

Demo records requirements

- The number of section records must provide enough data for nice-looking analytics (20 on average).
- The first three section records must be showcase records.
- You can add up to 1000 demo records to the corresponding database table.
- Record data must be logically consistent.
- Demo data must be relevant for the target audience and region where you are going to distribute the app (addresses, names, phone numbers).
- Each section record must be unique. The record variety must be reflected in folders and analytics.
- Use records that are logically related to records of other sections. We recommend creating record chains instead of going from section to section.
- Bind the demo records to base demo data for data integrity.
- Add demo records not only to sections, but also feed records, reminders, notifications, emails, comments,

likes, etc.

- Demo data must have a positive message.
- Demo data must be in English.
- You can fill out lookups using either softkey or demo data.
- Lookup records must be unique.
- If Creatio must add the corresponding activities when you add demo records, verify that the records are present in the activity schedule.
- If demo data must include dashboard analytics, verify the charts in each section and take the update date into account.
- Some sections use preconfigured filters by owner and period, for example [*Activities*], [*Invoices*]. Ensure the section list includes demo records that match the filter conditions.
- Implement a demo mode for connectors. You can do it using pre-configured test users or a simulated integration that showcase how the connector operates.

Learn more about the demo records in a separate article: [Demo version of the Marketplace app](#).