

# Create a new Syncope project



## Version warning

The content below is for Apache Syncope <= 1.2 - for later versions the [Getting Started](#) guide is available.

1. [Introduction](#)
2. [Prerequisites](#)
3. [Create new project](#)
  1. [Stable releases](#)
  2. [Snapshot \(development\) releases](#)
4. [Build new project](#)
5. [Where to go from here?](#)

## Introduction

This document describes how to create a new Syncope project.

A Syncope project includes (at least) two web applications: the **core** and the **console**. This page helps you get both web applications up and running with your own project as quickly as possible.

## Prerequisites

The preferred way to create a Syncope project is to generate a Maven project starting from published archetype.

Hence you need:

- Java SE Development Kit 6 (version 1.6.0-23 or higher) installed;
- [Apache Maven](#) (version 3.0.3 or higher) installed;
- Some basic knowledge about Maven;
- Some basic knowledge about [Maven WAR overlays](#);
- Some basic knowledge about [Maven archetypes](#).

## Create new project

Maven archetypes are templates of projects. Maven can generate a new project from such a template. For a project using Syncope, you need the website archetype. In the folder in which the new project folder should be created, type the command shown below. On Windows, run the command on a single line and leave out the line continuation characters ('\').

```
mvn archetype:generate \
-DarchetypeGroupId=org.apache.syncope \
-DarchetypeArtifactId=syncope-archetype \
-DarchetypeRepository=http://repo1.maven.org/maven2 \
-DarchetypeVersion=1.2.11
```

The archetype is configured with default values for all properties required by the archetype. If you want to customize any of these property values, type 'n' when prompted for confirmation.

You will be asked for:

1. the *groupId*  
something like 'com.mycompany'
2. the *artifactId*  
something like 'myproject'
3. the *version* number  
You can use the default; it is good practice to have 'SNAPSHOT' in the version number during development and the maven release plugin makes use of that string. But ensure to comply to the desired numbering scheme for your project.
4. the *package name*  
The java package name. A folder structure according to this name will be generated automatically; by default, equal to the groupId
5. (for archetypeVersion >= 1.0.5), the *secretKey*  
Provide any pseudo-random, 16 character length, string here that will be used in the generated project for AES ciphering.

Maven will create a project for you (in a newly created directory named after the value of the artifactId property you specified) containing two subprojects:

1. core - a pre-configured RESTful server, with JPA persistence
2. console - a web interface for dealing with the core

## Stable releases

Take a look at available releases of the archetype project at [central Maven repository](#), then change the `archetypeVersion` in the `mvn` command above accordingly.

## Snapshot (development) releases

As development go on, snapshot releases are published at [ASF repository](#).

If you want to test a snapshot release, be sure to:

1. change

```
mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate \
-DarchetypeGroupId=org.apache.syncope \
-DarchetypeArtifactId=syncope-archetype \
-DarchetypeRepository=http://repository.apache.org/content/repositories/snapshots \
-DarchetypeVersion=1.2.11-SNAPSHOT
```

2. add the following code right before `</project>` in root `pom.xml` of the generated project:

```
<repositories>
  <repository>
    <id>ASF</id>
    <url>https://repository.apache.org/content/repositories/snapshots/</url>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
</repositories>
```

## Build new project

Now build your project using the following command in the root folder of your project:

```
mvn clean package
```

This will produce two WAR files:

- `core/target/syncope.war`
- `console/target/syncope-console.war`

## Where to go from here?

Once you have create your project and verified that build went fine, you can now see how to [run an embedded demo](#) or [deploy to a real environment](#).