# From source


- Before you start
  - Software
  - Environment variables
- Checkout the project
- First time maven pre-build setup
  - Maven
    - 3rd party libraries  pre 3.3.0
    - First time maven build  pre 4.0.0
  - Basic build
  - Maven profiles and variables
    - Profiles
    - Variables
  - Environment specific builds
- Prepare database
  - Derby
  - MySQL
  - Other RDBMS's and configuring existing
- File paths configuration
  - Image repository
- Deploying
- IDE setup
- Import demo data


## Before you start

### Software

Before you start you must make sure that you have the following software is installed and properly configured (i.e. runnables are on system path and you can run them from command line).

| Software | Recommended version | Check runnable | Where to get it |
|---|---|---|---|
| Java SDK | 1.8.x+ | java -version | http://www.oracle.com/technetwork/java/javase/downloads/index.html |
| NodeJS | 8.x.x+  4.0.0<br>12.x.x+  4.1.0+ | node --version | https://nodejs.org/en/download/ |
| NPM | 6.4.x+  4.0.0<br>6.14.x+  4.1.0+ | npm -version | https://www.npmjs.com |
| Maven | v.3.0+ | mvn -version | http://maven.apache.org/download.html |
| Ant | v.1.8+ | ant -version | http://ant.apache.org/bindownload.cgi |
| Ant contrib | v.1.0b+ | | http://ant-contrib.sourceforge.net |
| Tomcat | v.7+ | | http://tomcat.apache.org |
| MySQL | v.5.5+ | mysql | http://dev.mysql.com/downloads/ |

If you are running ***nix** please consult your OS documentation for the installation procedures of the above software. We would not recommend installing it manually. For example Mac OSX users are strongly advised to you macports as installation shell.

| Nice to have | Notes |
|---|---|
| Intellij IDEA Ultimate | Ultimate is required for Flex support |
| Http server configurable for ajp 1.3 | If you would like to mirror production setup locally |
| TOAD or MySQL query browser | If you plan to use mysql then GUI client would help |

## Environment variables

| Environment Variables | Required | Example value | Notes |
|---|---|---|---|
| JAVA_HOME | ✅ | D:\jdk1.8.0_45 | |
| JAVA_OPTS | | -Xms256m -Xmx1024m | Required as Tomcat may fail with OutOfMemory exception |
| ANT_HOME | ✅ | D:\apache-ant-1.8.1 | |
| ANT_OPTS | | -Xms256m -Xmx1024m | |
| M2_HOME | ✅ | D:\apache-maven-3.0.3 | |
| MAVEN_OPTS | ✅ | -Xms256m -Xmx1024m | Required as test may fail with OutOfMemory exception |
| CATALINA_OPTS | | -Dfile.encoding=UTF-8 | Required by Tomcat to correctly process UTF-8 requests |
| PATH | | | should contain all executables for the installed software |

> You can check Environment variables from shell (*nix) by using "./yc.sh env" or cmd (Windows) by using "yc.bat env".

# Checkout the project

The repository is located at GutHub https://github.com/inspire-software/yes-cart.
Please refer to instruction on GutHub https://help.github.com/articles/which-remote-url-should-i-use/ on the cloning/forking options available.

> Please **do not use spaces in directory names** where you checkout the project. This may prevent you from accessing some of useful script tools and delay workspace setup.

# First time maven pre-build setup

> You only need to perform steps in this section **once**. After these are correctly setup skip this section in day to day builds.

## Maven

### 3rd party libraries  pre 3.3.0

YesCart uses some 3rd party libraries that are no available on maven central, so you need to manually add them to your local repository

Go to the root of the YesCart checkout directory (i.e. /dev/yc in the example below).

```
$ cd /dev/yc
```

If you are running ***nix** you can use this nifty script

```
/dev/yc$ ./yc.sh i3rd
```

If you are running **Windows** use batch file

```
C:\yc\> yc.bat i3rd
```

> If above does not work please make sure that your mvn installation is working correctly from command line and that all environment variables have been setup correctly.

### First time maven build  pre 4.0.0

Run maven *clean* install command that will compile, test and package the code using the *dev* environment configurations

> Note the clean phase in the first time build. It was necessary in pre 4.0.0 versions to load all nodes dependencies for the jam-jsclient module.

```
/dev/yc$ mvn clean install
```

# Building using maven

> If build is breaking make sure you have setup all that is necessary from "First time maven pre-build setup" section above.

### Basic build

To build the project and deploy it to Tomcat all you need is to execute couple of commands in shell/cmd.

Go to the root of the YesCart checkout directory (i.e. /dev/yc in the example below).

```
$ cd /dev/yc
```

Run maven install command that will compile, test and package the code

```
/dev/yc$ mvn install
```

This command will perform default build which is targeted at **development** environment, which should be suitable in most cases.

As maven will be processing each module you will be able to see current configuration in echo along with some tips as to what other options are available.

**Example echo configurations**

```
     [echo]
================================================================================
================================================================================
=====
     [echo] |
     [echo] | __     _______                            ______
     [echo] | \ \   / / ____|                          |  ____|
v.3.5.0-SNAPSHOT (build: 0556dbaf)
     [echo] |  \ \_/ | |   ______ _ __  _   _ _ __ ___| |__   ___ ___  _ __
___  _ __ ___   ___ _ __ ___ ___
     [echo] |   \   /| |  |______| '_ \| | | | '__/ _ |  __| / __/ _ \| '_
` _ \| '_ ` _ \ / _ | '__/ __/ _ \
     [echo] |    | | | |____     | |_) | |_| | | |  __| |___| (_| (_) | | |
| | | | | | | |  __| | | (_|  __/
     [echo] |    |_|  \_____|    | .__/ \__,_|_|  \___|______\___\___/|_|
|_| |_|_| |_| |_|\___|_|  \___\___|
     [echo] |                    | |
     [echo] |                    |_|  Copyright 2018
http://www.yes-cart.org
     [echo] |
     [echo]
================================================================================
================================================================================
=====
     [echo]
     [echo] YC module jam-3.5.0-SNAPSHOT-rev.0556dbaf configurations:
     [echo]
     [echo] Environment configuration (use -Denv=x to switch)
               : dev-idea
     [echo] Database provider (use -Pmysql or -Pderby)
                     : derby
     [echo] Full text search provider (use -PftEmbededLucene or
-PftDisabled) : embeded-lucene
     [echo] Tomcat SSL configuration (use -Pssl)
                            : off
     [echo] JAM npm goal (use -Dnpm.goal=x to switch)
```

```
                        : build.dev
     [echo] JAM npm target (use -Dnpm.target.home=x to switch)
                : dev (typescript/dist/dev)
     [echo] JAM npm modules (use -Dnpm.target.modules=x to switch)
           : node_modules (typescript/node_modules)
     [echo]
     [echo] Payment modules
     [echo] LiqPay payments (use -PpaymentLiqPay or -PpaymentAll)
            : on
     [echo] Cybersource payments (use -PpaymentCybersource or -PpaymentAll)
  : on
     [echo] Authorize.Net payments (use -PpaymentAuthorize or -PpaymentAll)
  : on
     [echo] PayPal payments (use -PpaymentPaypal or -PpaymentAll)
            : on
     [echo] PostFinance payments (use -PpaymentPostFinance or -PpaymentAll)
  : on
     [echo]
     [echo] Add-on modules
     [echo] Marketing price rules integration  (use -Ppricerules)
            : on
     [echo]
     [echo] Web Apps
     [echo] Storefront: ROOT.war
     [echo] REST API  : yes-api.war
     [echo] ADMIN     : yes-manager.war
     [echo]
     [echo] Tips:
     [echo] * use mvn help:all-profiles or help:active-profiles to list
profiles available
     [echo] * use mvn dependency:tree to list all dependencies
     [echo] * documentation for YC profiles is available at:
     [echo]
http://www.inspire-software.com/documentation/wiki/docyescart/view/3.3.x/F
or%20Technical%20Users/Installation/From%20source/
     [echo]
```

```
================================================================================
================================================================================
=====
```

**mvn profiles**

The platform comes with a variety of maven profiles to tweak your build as required

# Maven profiles and variables

## Profiles

When working with maven if you specify at least one profile then activation by default stops working for other profiles. Hence you may need to specify all default profiles in some cases.

| Profile | Alternative | Default | Dependency [1] | Affected module | Version | Notes |
|---|---|---|---|---|---|---|
| dev | | | | jam, sf, api | 3.0.0+ | synonym for -Denv=dev |
| devIntellijIDEA | | | | jam, sf, api | 3.0.0+ | synonym for -Denv=dev-idea |
| ~~buildInstaller~~ (**EoL 3.3.x**) | | | | installer | 3.0.0+ | Includes installer module in the build |
| derby | mysql, mysql8 | ✅ | ✅ | jam, sf, api | 3.0.0+ | Sets derby connector for hibernate |
| mysql | derby, mysql8 | | ✅ | jam, sf, api | 3.0.0+ | Sets mysql connector for hibernate (connector/j 5.x. |
| mysql8 | mysql, derby | | ✅ | jam, sf, api | 4.0.0+ | Sets mysql connector for hibernate (connector/j 8.x |
| ftEmbededLucene | ftDisabled | ✅ | ✅ | sf, api | 3.5.0+ | Enables Lucene to be full text search provider for st |
| ftDisabled | ftEmbededLucene | | | sf, api | 3.5.0+ | NOOP full text search provider for storefront (used f provider) |
| ~~nodejs~~ ⚠ (Deprecated 4.0.0) | | | | jam-jsclient | 3.3.0+<br>~~4.0.0+~~ | ❌ Deprecated in 4.0.0, please use admincors and<br>⚠ Used for JAM client development with validate is from within specific module:<br>`yes-cart/manager/jam$ -Pdev,nodejs,derby,ftE`<br>⚠ Use this profile for supporting IDE builds to cop |
| ssl | | | | jam, sf, api | 3.0.0+ | Sets secure configurations for wicket (enabled secu default secure and insecure ports (8443, 8080). If n (disables security, guarantee NONE), default secur |
| cors | corsOff | ✅ | | api | 3.7.0+ | Allows to enable CORS mode for API. See REST A |
| corsOff | cors | | | api | | Disable CORS mode for API (original 3.6.x behavio |

| | | | | | | |
|---|---|---|---|---|---|---|
| admincors | admincorsOff | | | jam | 4.0.0+ | Enables CORS in Admin app APIs. It is recommend that SPA started in npm can connect to the locally r script **startlocal.sh** located in jam-jsclient/src/main/ which will connect to APIs on localhost:8082/cp (wh |
| admincorsOff | admincors | ✓ | | jam | 4.0.0+ | Disables CORS in Admin app APIs. |
| templates ⚠ | | | | jam, sf, api | 3.0.0+ | ⚠ DO NOT use for full builds. This profile allows relevant module. Usage is from within specific modu<br>`yes-cart/manager/jam$ -Ptemplates,derby,ftEm`<br>⚠ Use this profile for supporting IDE builds to cop |
| connREST | connWS, connJGIPv4, connJGIPv6 | ✓ | ✓ | jam, sf, api | 3.6.0+ | REST API connector for internal communication be |
| connWS | connREST, connJGIPv4, connJGIPv6 | | ✓ | jam, sf, api | 3.6.0+ | WebServices connector for internal communication |
| connJGIPv4 | connREST, connWS, connJGIPv4 | | ✓ | jam, sf, api | 3.6.0+ | JGroups IPv4 connector for internal communication |
| connJGIPv6 | connREST, connWS, connJGIPv6 | | ✓ | jam, sf, api | 3.6.0+ | JGroups IPv6 connector for internal communication |
| paymentBase | paymentAll | ✓ | ✓ | payment modules | 3.0.0+ | Disabled all additional payment modules |
| paymentAll | paymentBase | | ✓ | payment modules | 3.0.0+ | Enables all additional payment modules |
| paymentLiqPay | paymentLiqPayOff | | | payment modules | 3.0.0+ | Enables LiqPay additional payment module |
| paymentLiqPayOff | paymentLiqPay | | | payment modules | 3.0.0+ | Disables LiqPay additional payment module |
| paymentCybersource | paymentCybersourceOff | | ✓ | payment modules | 3.0.0+ | Enables Cybersource additional payment module |
| paymentCybersourceOff | paymentCybersource | | | payment modules | 3.0.0+ | Disables Cybersource additional payment module |
| paymentAuthorize | paymentAuthorizeOff | | ✓ | payment modules | 3.0.0+ | Enables Authorize.NET additional payment module |
| paymentAuthorizeOff | paymentAuthorize | | | payment modules | 3.0.0+ | Disables Authorize.NET additional payment module |
| paymentPaypal | paymentPaypalOff | | ✓ | payment modules | 3.0.0+ | Enables PayPal additional payment module (PayPa |
| paymentPaypalOff | paymentPaypal | | | payment modules | 3.0.0+ | Disables PayPal additional payment module (PayPa |
| paymentSwissbilling | paymentSwissbillingOff | | ✓ | payment modules | 3.7.0+ | Enables Swissbilling additional payment module |
| paymentSwissbillingOff | paymentSwissbilling | | | payment modules | 3.7.0+ | Disables Swissbilling additional payment module |

1 Dependency profiles are used to inject additional Maven dependencies during build. **If at least one Maven profile is specified in build command all applicable dependency profiles must be explicitly stated**. For example "mysql" profile adds MySQL driver jar file to WEB-INF/lib thus if we use "-PdevIntellijIDEA" we need to use either "-PdevIntellijIDEA,mysql" or "-PdevIntellijIDEA,derby" to add either MSQL or Derby driver. **Note that all jar files are preserved between builds in WEB-INF/lib unless a "mvn clean install" is performed**.

## Variables

The platform defines several build variables in pom files. However because of the dependency management mechanism using only variable is not enough. Thus you must not manipulate those manually but rather define correct values in custom profiles within pom.xml files.

## Environment specific builds

As mentioned in the "Basic build" section above, by default development configuration is used. This configuration is defined by **env** variable.

This variable specified which configuration location to use in: /dev/yc/maven directory.

Each configuration is grouped into a directory. For example development is located in "dev" and special configuration for Intellij IDEA in "dev-idea" etc.

Custom configuration for other environments can be crafted to your specification and controlled through **env** variable.

For example for continuous integration server you may run the following maven build:

```
/dev/yc$ mvn install -Pci
```

Which would read configurations from:

```
/dev/yc/maven/ci
```

This gives a very flexible and simple mechanism to configure environment specific builds.

# Prepare database

Default setup provides two choices: **derby** to use for development environments, or use **MySQL** server. The connection configurations are defined in the environment specific configuration files. However because RDMBS requires specific driver relevant profiles must be used to include driver dependency in the build.

> If you want to use other RDBMS or change some of the configuration you will need to either change existing maven profile or add a new profile.

## Derby

> If you encounter **java.security.AccessControlException: access denied ("java.net.SocketPermission" "localhost:1527" "listen,resolve")** refer to Derby quick startguide.

Navigate to YC_HOME from command line and startup Derby server (in background mode)

***nix**

```
/dev/yc$ ./yc.sh derbygob
```

**Windows**

```
C:\yc\> yc.bat derbygob
```

Run the database initialisation script

***nix**

```
/dev/yc$ ./yc.sh dbiderby
```

**Windows**

```
C:\yc\> yc.bat dbiderby
```

Follow instructions to complete the setup.

> ***nix**
> Unix users can use grep to quickly check the derby log for errors

```
grep -i 'error' ./derbyinit.log
```

> If you experience problems, try to setup databases manually.

## MySQL

Make sure your MySQL server is running (try connecting to it through command line 'mysql' command)

Navigate to YC_HOME from another command line and execute

***nix**

```
/dev/yc$ ./yc.sh dbimysql
```

**Windows**

```
C:\yc\> yc.bat dbimysql
```

Follow instructions to complete the setup.

If you experience problems, try to setup databases manually.

## Other RDBMS's and configuring existing

The hibernate connection properties are defined in environment specific configuration files in **YC_HOME/env/maven**, which you can add or modify. YesCart provides derby and mysql profiles by default, which are located in main POM in YC_HOME/pom.xml.

Profiles look like this:

**MySQL profile example**

```
...
       <!-- Variable definition in main pom.xml that allows to include
relevant configuration file through maven filters -->
       <profile>
           <id>mysql</id>
           <properties>
               <env.db>mysql</env.db>
           </properties>
       </profile>
...
       <!-- Module specific dependencies (drivers) -->
       <profile>
           <id>mysql</id>
           <dependencies>
               <dependency>
                   <groupId>mysql</groupId>
                   <artifactId>mysql-connector-java</artifactId>
                   <version>${mysql-connector.version}</version>
               </dependency>
           </dependencies>
       </profile>
...
```

**mysql** profile above sets variable **env.db** to **mysql** which causes maven to pick up config-db-mysql.properties file and then insert provided configurations into META-INF/context.xml for all web apps.

To add your own RDMBS profile just follow the same principle (e.g. env.db=oracle) and then create config-db-oracle.properties which would have definition of all configuration properties. Then in web app modules add dependency to the necessary driver dependencies

# File paths configuration

## Image repository

Image repository is set via system preferences (SYSTEM_IMAGE_VAULT). Default locations are defined in the environment specific file yc-config.properties in the following properties:

```
imagevault.default=file://${basedir}/../../theme/imagevault/src/main/resou
rces/default/imagevault/
filevault.default=file://${basedir}/../../theme/imagevault/src/main/resour
ces/default/filevault/
sysfilevault.default=file://${basedir}/../../theme/imagevault/src/main/res
ources/default/sysfilevault/
```

This default configuration points to the theme module default directory layout in the root of your project. We highly recommend setting this to directory outside of your code base. Note that this directory is shared by all YC web apps

# Deploying

Once maven build is completed several JAR/WAR files will be built in target directories of each module.

You need to **copy the shop manager, storefront and REST API war files into the $TOMCAT_HOME/webapps**

```
/dev/yc$ cp ./manager/jam/target/yes-manager.war $TOMCAT_HOME/webapps
/dev/yc$ cp ./web/store-wicket/target/ROOT.war $TOMCAT_HOME/webapps
/dev/yc$ cp ./web/api/target/yes-api.war $TOMCAT_HOME/webapps
```

If you are using **mysql** you may have problems during tomcat startup because of mysql-connector. The symptom is **javax.naming.NamingException: com.mysql.jdbc.Driver** in your catalina.out log. In order to resolve this copy /yes-shop/WEB-INF/lib/mysql-connector-java-5.1.13.jar to share lib directory of your tomcat

Make sure your SQL server is running and **start the tomcat instance** as you would normally either through services, catalina.bat or catalina.sh depending on the type of installation of your tomcat instance.

If you are setting up tomcat from scratch please have a look at some things on our Tomcat page that you should be aware of.

Open your browser and navigate to the following url for the **storefront**:

> *http://localhost:8080/*

Or open your browser and navigate to the following url for the **manager application**:

> *https://localhost:8443/yes-manager/*

Default user/password to connect is:

| | |
|---|---|
| username | admin@yes-cart.com |
| password | 1234567 |

Or use REST client to access **API**:

> *http://localhost:8080/yes-api/*

# IDE setup

Intellij IDEA

# Import demo data

Follow steps on Demo Data Import