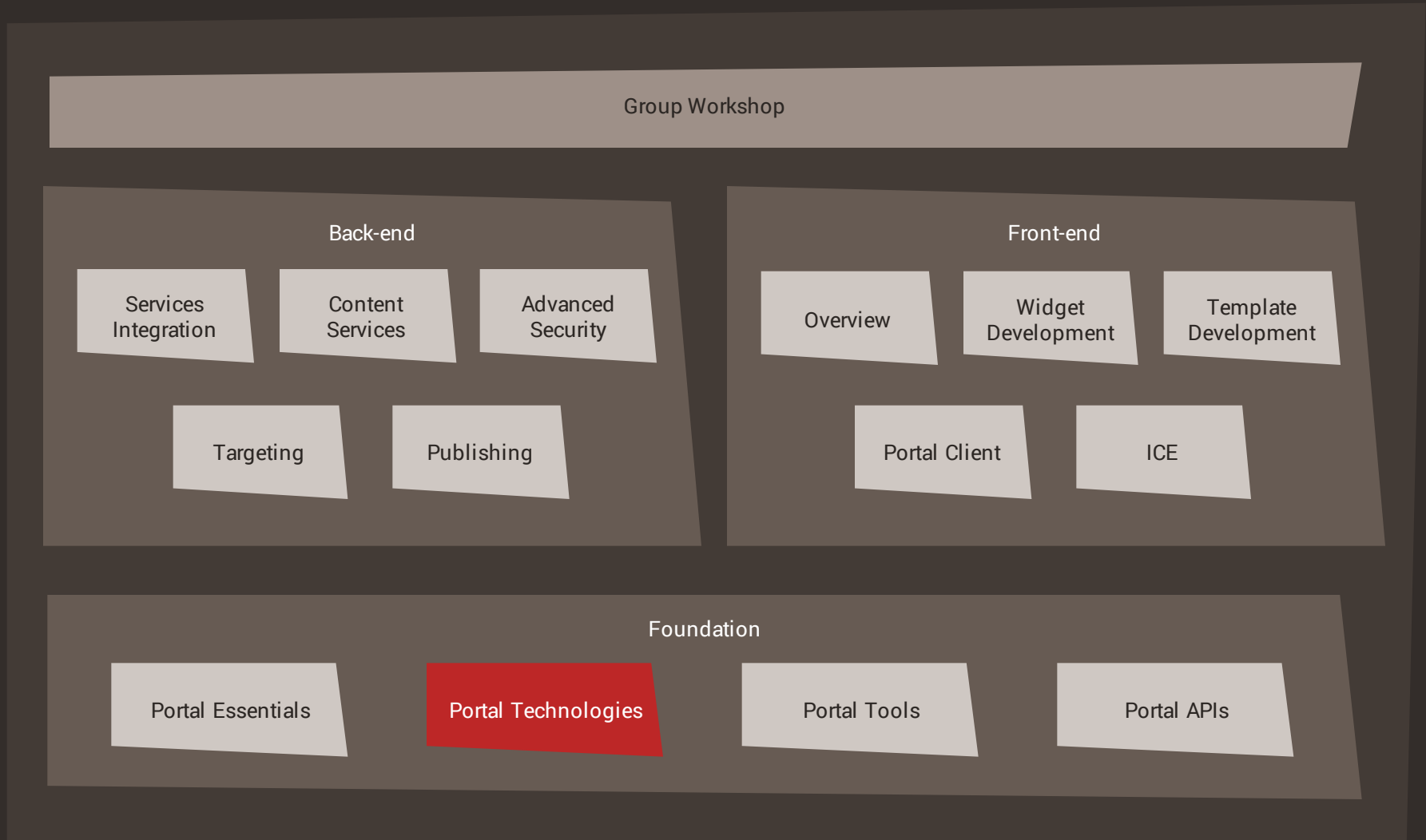


Portal Technologies

Focus Area: Foundation

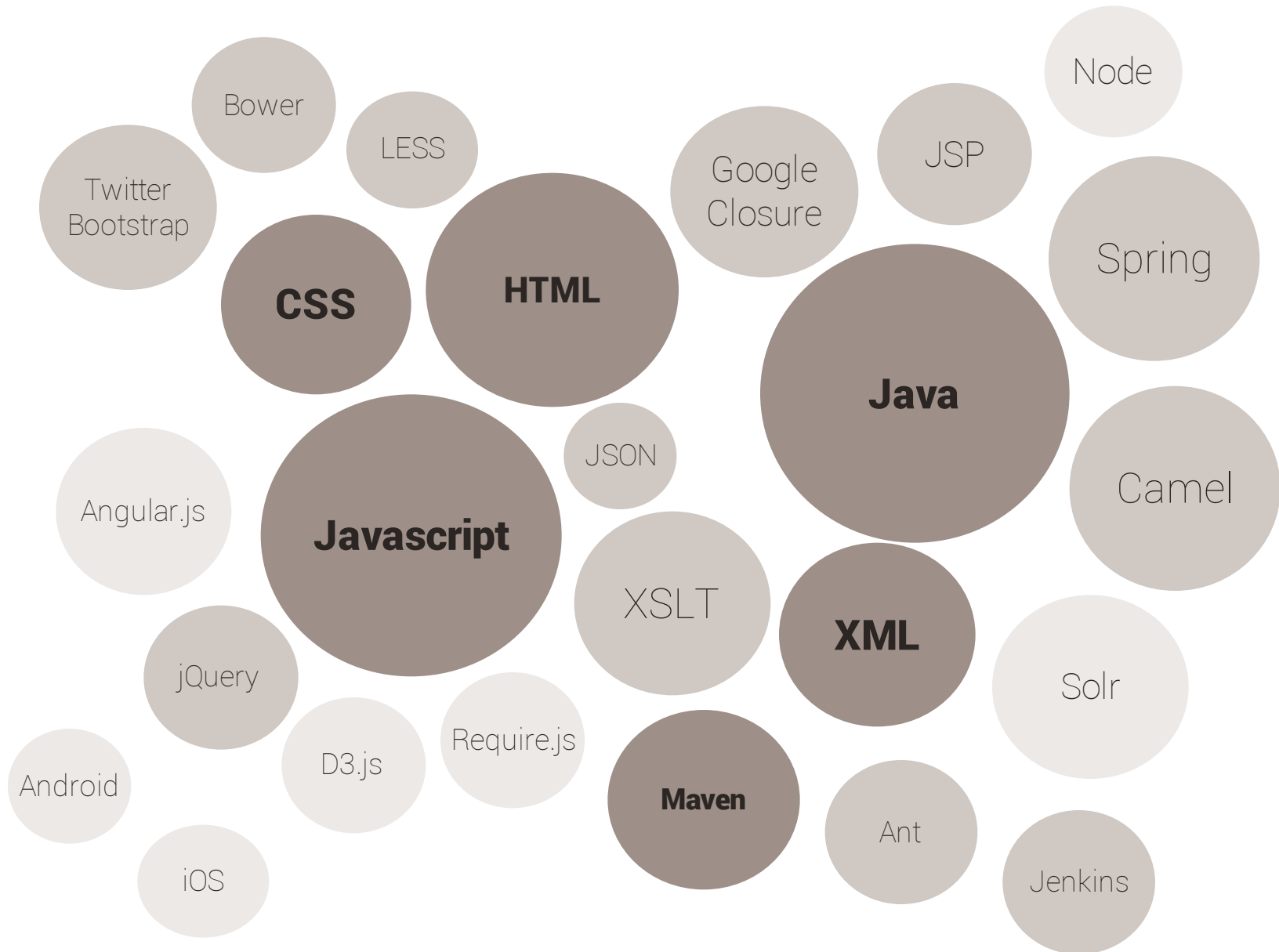




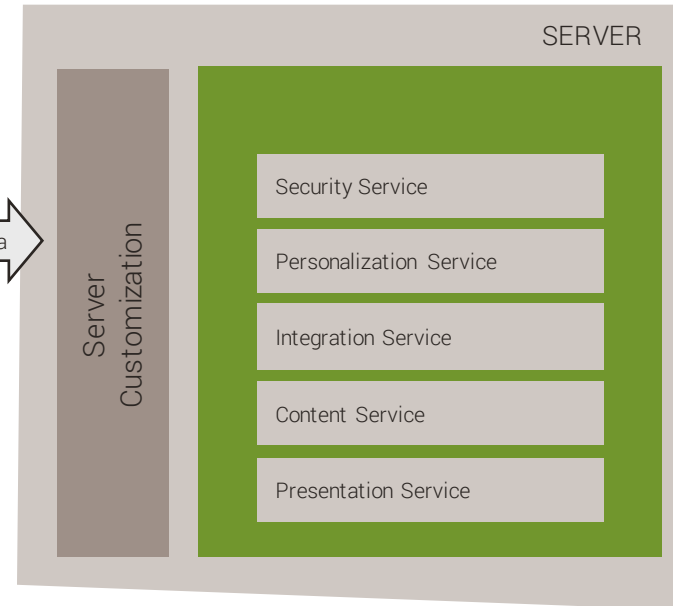
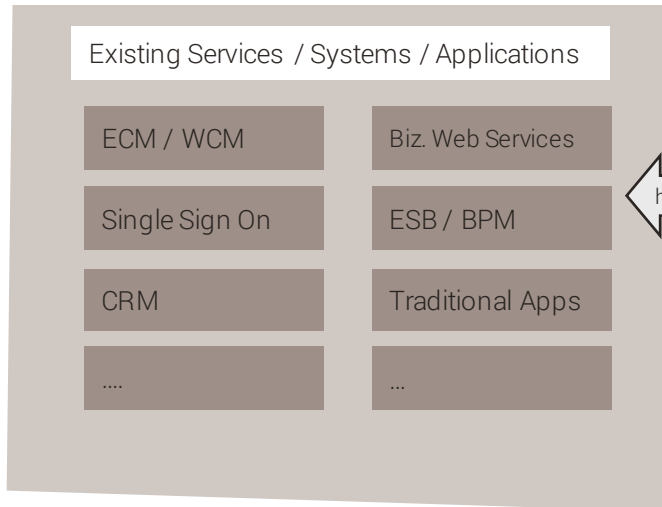
1. Get to know the technologies Portal is using under the hood
2. Get to know how to install and run a Portal server in your development environment
3. Become familiar with the development setup's directories and files

Languages and Standards

Portal Technologies



Client side developer



Existing applications developer



Java backend developer



Technologies Overview

Portal Technologies

- Portal is an Enterprise Java Application
- Makes heavy use of Spring under the hood
- Key area's of Spring
 - Spring IOC
 - Spring MVC
 - Spring Security

- Portal client can render the model client side with JS
- Widget behavior is usually written with JS
- jQuery is shipped with Portal as a Portal Client dependency
 - But not a widget dependency

- Very ubiquitous throughout Portal
- Most configuration
 - Web app
 - Spring
 - Maven
- REST data interchange format

- Software tool for software project management and build automation.
- Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.
- Convention over configuration
 - `/src/main/java`
 - `/src/main/webapp`
 - `/src/test`

- Archetype is a Maven project templating toolkit
 - An original pattern or model from which all other things of the same kind are made.
- Artifacts
 - An artifact is a file, (usually a JAR), that gets deployed to a Maven repository.
 - Each artifact has a group ID (usually a reversed domain name, like com.example.foo), an artifact ID (just a name), and a version string.
 - A project's dependencies are specified as artifacts.
 - Artifacts are stored in a repository
 - Central repository
 - Local repository

- The Project Object Model (`pom.xml`)
 - Project meta data
 - Dependencies
 - Build lifecycle
 - Plugins

- Build lifecycle

<code>process-resources</code>	<code>resources:resources</code>
<code>compile</code>	<code>compiler:compile</code>
<code>process-test-resources</code>	<code>resources:testResources</code>
<code>test-compile</code>	<code>compiler:testCompile</code>
<code>test</code>	<code>surefire:test</code>
<code>package</code>	<code>jar:jar</code>
<code>install</code>	<code>install:install</code>
<code>deploy</code>	<code>deploy:deploy</code>

- Backbase hosts a repository that provides a convenient mechanism to setup a local Maven development environment.
- <https://repo.backbase.com/repo>
- Archetypes available from this repository:
 - Backbase Portal Archetype Blank
 - Launchpad Archetype

- Jetty is lightweight HTTP server and servlet container

- `maven-jetty-plugin`

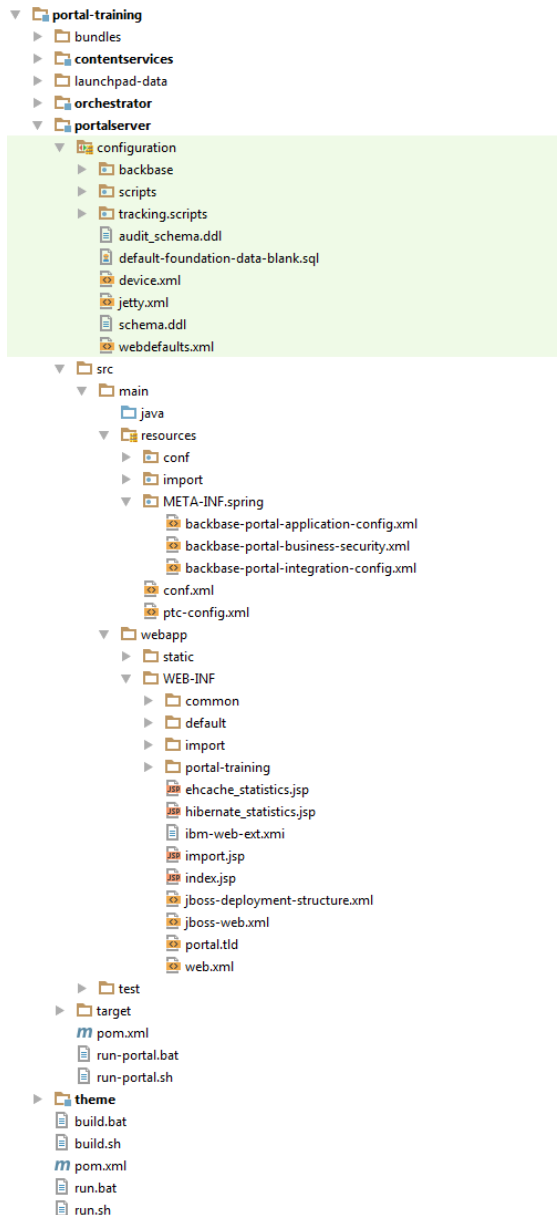
- Run using command

`mvn jetty:run`

- If you want your data to be reset to initial data then before `jetty:run` first execute the command:

`mvn clean install` (blank archetype)

`mvn clean -Pclean-database` (launchpad archetype)



Highlights

- Application configuration
- Jetty configuration
- Security configuration
- Database
- pom.xml
- web.xml

What you do not see in the explorer

- Mashup config
- Content Services config
- URL access control config

```

v [icon] lp56-mobile
  > [icon] configuration
  > [icon] exp-exports
  > [icon] data
  > [icon] dist
  > [icon] parent
  > [icon] services
  v [icon] statics
    > [icon] collection
    > [icon] dist
    [icon] build.bat
    [icon] build.sh
    [icon] pom.xml
    [icon] statics.iml
  > [icon] styleguide
  > [icon] tools
  v [icon] webapps
    > [icon] contentservices
    > [icon] orchestrator
    v [icon] portalserver
      v [icon] src
        > [icon] demo
        v [icon] main
          > [icon] resources
          v [icon] webapp
            > [icon] static
            > [icon] WEB-INF
        > [icon] test
      > [icon] target
    [icon] build.bat
    [icon] build.sh
    [icon] initial_build.bat
  
```

Highlights

- **webapps/** : Portal, CS & Orchestrator
- **configuration/** : all Backbase-specific configuration
- **services/** : custom Camel services
- **statics/** : Launchpad collection & custom items
- **styleguide/** : documentation of Launchpad Components

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-ns"
  xsi:schemaLocation="
    http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.backbase.training</groupId>
  <artifactId>blank</artifactId>
  <version>1</version>
  <packaging>war</packaging>

  <name>blank</name>

  <repositories>[]
  <pluginRepositories>[]
  <properties>[]
  <dependencies>[]
  <build>[]
</project>
```



Follow the “Getting Your First Launchpad-Based Portal Set Up” how to guide to create your own developer setup.

<https://my.backbase.com/resources/how-to-guides/getting-your-first-launchpad-based-portal-set-up/>

Archetype Structure

Portal Technologies

- Configuration: `/configuration`
- Source and test: `/src/main`, `/src/test`
 - Resources and config: `/src/main/resources`
 - Spring config: `/src/main/resources/META-INF/spring`
 - Portal import scripts: `/src/main/resources/import`
 - Java code: `/src/main/java`
 - Webapp code: `/src/main/webapp`
 - Static files: `/src/main/webapp/static`
 - WEB-INF: `/src/main/webapp/WEB-INF`
- Maven build directory: `/target`

- POM (Project Object Model): `/pom.xml`
 - Dependencies
 - Build
 - Plugins
 - Jetty
- Web app config: `/src/main/webapp/WEB-INF/web.xml`
 - Links to Spring config

Bundles

Portal Technologies

- A bundle is project module that is developed outside of the portal server project.
- It can have an independent Software Development Life Cycle (SDLC) and can be shared by multiple portal projects.
- Packaging your extensions as bundles outside of portal server is a best practice.

- **Separation of concern** – An individual module that has little or no dependencies.
- **Flexibility in releasing** – Bundles can have their own life cycle so that one project is not delayed by another.
- **Isolation in testing** – Every bundle can be tested without waiting for others, or worry about impacting other bundles.
- **Automation** – We use Maven as a build tool for both Java and static resources to configure automation of packaging and deploying while minimizing the risk of human error.
- **Increased frequency of delivery** – If static resources are built outside of the portal server directory, adding, releasing, and updating of new widgets, containers, and/or themes can be done without taking the portal server offline.



Follow the “What Are Bundles?
How Can They Help Me? And How
Do I Make Them?” how to guide to
create your own bundle.

[https://my.backbase.com/resources/how-to-guides/
what-are-bundles-how-can-they-help-me-
and-how-do-i-make-them/](https://my.backbase.com/resources/how-to-guides/what-are-bundles-how-can-they-help-me-and-how-do-i-make-them/)

Collections

Portal Technologies

Collections are a convenient way of grouping custom items together (widgets, containers, templates, chromes, etc...).

A collection is simply a maven project, that you can add as a dependency of your portal. It can be developed, tested and deployed independently from your portal project. It can also be re-used in multiple projects.

The contents of a collection is managed by Bower. At the root of each collection, you will find a `bower.json` file, listing the available items, for example:

```
{
  "name": "my-custom-collection",
  "version": "0.1.0",
  "private": true,

  "dependencies": {
    "chrome-templates": "^1.0.1",
    "container-column": "^1.0.1",
    "container-panel": "^1.0.1",
    "container-tabs": "^1.0.1",
    "page-launchpad-base": "^1.0.3",
    "widget-device-dna": "^1.2.2",
    "widget-login": "^2.2.3",
    "widget-navbar": "^3.0.1",
    "widget-widgets-catalog": "^1.1.2",
    "template-launchpad-page": "^1.0.1"
  }
}
```

- Portal uses a core set of technologies,
but can integrate with many more
- Maven is a convenient way to run
a local Portal development
environment

Thank you!

www.backbase.com
sales-eu@backbase.com

New York: +1 646 478 7538
Amsterdam: +31 20 465 8888