

Ebook Ecommerce

Este documento tem como objetivo explicar de maneira sucinta a solução adotada para construção da ebook commerce.

Requisitos

- Java 1.6 ou mais recente
- MySql 5.x
- Maven 3.x
- Conexão com a internet
- Código Fonte
 - Github: <https://github.com/camilolopes/workspaceAngularJs>
(ebookcommerce-parent)
 - Download war/Documento: <http://camilolopes.com.br/avenuecode/>

Frameworks /API

- Spring core 3.x
- Hibernate 4.x
- Jersey
- AngularJS 1.x
- DBUnit
- REST
- easyJavaDropboxAPI
- Flyway
- bootstrap

[Flyway](#)

É um plugin que tenho utilizado junto com o Jenkins no ambiente de integração continua para banco de dados. Utilizei no projeto para a criação das tabelas e massa de dados serem carregados de maneira simples, rápida e fácil.

[easyJavaDropbox API](#)

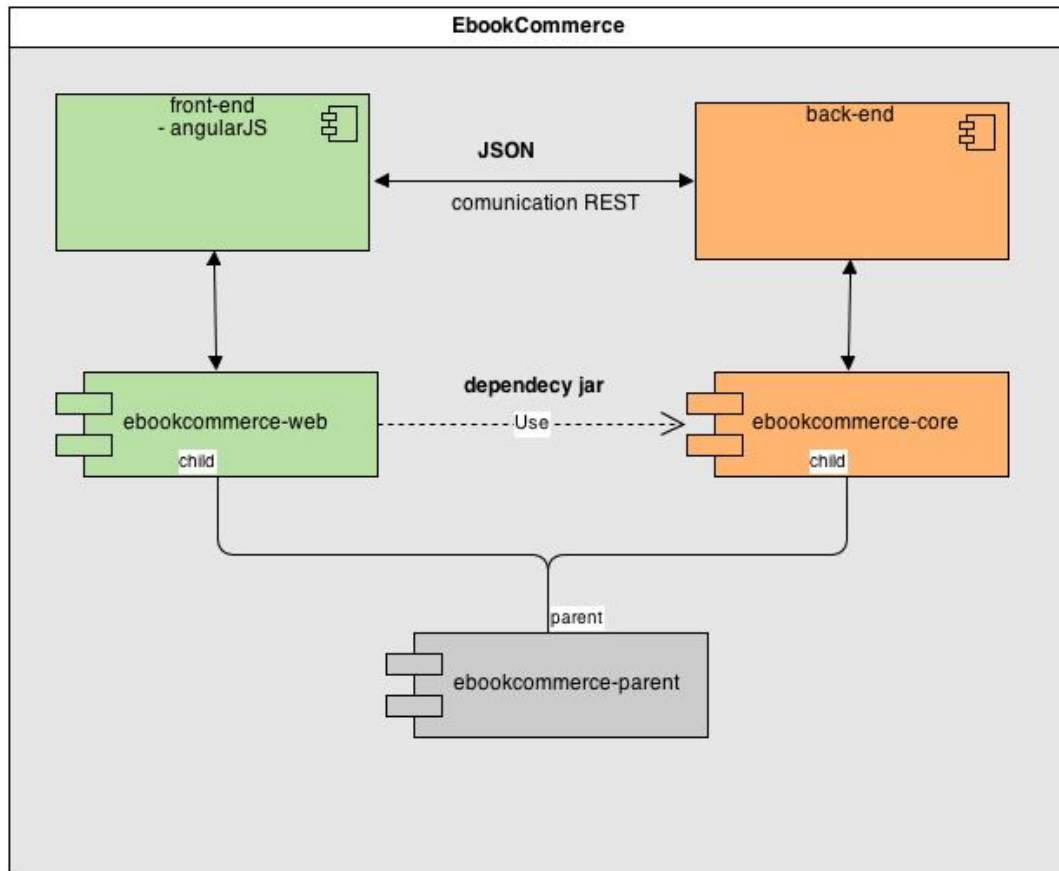
É uma API que desenvolvi alguns meses atrás para comunicação com o dropbox em Java no desenvolvimento de aplicações enterprise. Para um projeto mais recente que trabalhei a API foi usada para gerenciamento de imagens em uma loja, tendo como objetivo de que as imagens ficassem na cloud e atualização fosse de maneira transparente e simples. Na prática se precisamos atualizar a imagem de um ebook/produto basta realizar um update no dropbox e essa atualização é refletida na loja. A partir dessa API nasceu [uma versão web](#) feita em angularJS para geração de token de comunicação com Dropbox. Reutilizei essa arquitetura que desenvolvi na construção da ebook commerce.

Note Importante:

- Não foi usado unit test para camada de front end;
 - Razão: conhecimento limitado em unit test na camada de front end com JS.
- As imagens do produto vem do dropbox através da API [easyJavaDropbox API](#);
 - Questões de segurança e acesso não foram tratadas no exemplo ebook commerce.

- Aplicação não suporta <= IE8 devido a limitação do angularJS.
- Detalhes de mensagem e interação com o usuário não foram tratados na loja;
- O front-end para cadastro dos produtos não foi possível desenvolver há tempo. Sendo assim, a massa de dados será persistida através de arquivos .sql.
- Shopping Cart: foi usado o recurso de local storage, para essa funcionalidade realizei pesquisas na internet, grupos e fóruns do angularJS, pois possuía um conhecimento limitado em local storage com angularJS.

Application Solution



- O projeto foi dividido em módulos com objetivo de facilitar a manutenção e separação de responsabilidades. Uma das vantagens dessa arquitetura que o front-end e back-end não são dependentes e alteração de framework no front-end (ex.: de angularJS para Alloy) ou alteração de linguagem no back-end(de Java para Ruby) não impactaria em uma mudança em todo o projeto. Comunicação front-end & back-end é feita via RESTful usando objetos JSON. O back-end pode ser reutilizado em criação de uma outra loja onde o angularJS não possa ser usado.
 - Ebookcommerce-web: é o modulo responsável por cuidar do front-end da loja
 - Ebookcommerce-core: é todo back-end da loja, desenvolvido em Java. É nesse modulo que tratamos unit test do lado do servidor, regras de negócio etc.
 - Ebookcommerce-parent: é o modulo pai que gerencia os dois outros módulos;

Acessando Online

Disponibilizei uma versão live:

<http://apps.camilolopes.com.br/ebookcommerce>

Rodando Localmente

Step 1

1. Crie o schema camilo_ebook no MySql 5.x
2. Crie o usuário camilolopes com a senha camilo2593

Note: caso deseje usar outras informações para o banco de dados, é preciso alterar o arquivo app-context.xml no modulo ebookcommerce-core e realizar as alterações nos arquivos .sql (/ebookcommerce-core/src/main/resources/db/migration). Altere também as informações no ebookcommerce-core/pom.xml para o plugin do flyway com as informações do banco de dados.

Step 2

Após configurado o banco de dados, para criação das tabelas e massa de dados basta rodar os seguinte comandos no ebookcommerce-core

1. mvn clean compile flyway:info
2. mvn flyway:migrate

Step 3

1. via command line vá até o modulo ebookcommerce-parent
2. execute mvn clean install

Step 4

1. subindo aplicação mvn tomcat:run

<http://localhost:8080/ebookcommerce/>