Hightide Documentation

Release 1.0.0

Ground Zero Labs

March 31, 2014

Contents

CHAPTER 1

Introduction to Hightide

Getting Started

2.1 Installing Hightide

2.1.1 Requirements

You need Java SDK v1.8 to run Hightide. Check your current java installation:

```
$ java -version
```

2.1.2 Manual Installation

You can manually download Hightide's distribution and extract it on your preferred directory. You will need to setup `HIGHTIDE_HOME` environment variable and add distibution's `/bin` directory to your `PATH` variable.

2.1.3 Homebrew Installation [Mac OS X]

For mac users you can also install Hightide using Homebrew:

```
$ brew install hightide
```

2.2 Creating your first application

Once you're done with the installation, check if Hightide is successfully installed using the following command:

```
$ hightide
```

You should see something like the following output:

```
    __  ___       __    __  _     __
   / / / (_)___ _/ /_  / /_(_)___/ /__
  / /_/ / / __ `/ __ \/ __/ / __  / _ \
 / __  / / /_/ / / / / /_/ / /_/ /  __/
/_/ /_/_/\__, /_/ /_/\__/_/\__,_/\___/
        /____/


hightide>
```

You can now create a new Hightide application from Hightide's shell prompt:

```
hightide> new mynewapp
```

or straight from your command line prompt:

```
$ hightide new mynewapp
```

> **NOTE:** All Hightide commands can be executed either from Hightide's shell or command line [1]. More information on Hightide Shell available *here*

The above command will create a new directory named `mynewapp` and all necessary files to run a Hightide application, using the default Hightide prototype.

Application prototypes is a very usefull concept and the default way of making a new application in Hightide.

[1] Except the `exit` command!

CHAPTER 3

Using the command line

..at least if there's nothing better to use!

CHAPTER 4

Boilerplate-free code

CHAPTER 5

Configuring Hightide application

CHAPTER 6

Routing application requests

CHAPTER 7

Working with Resources

Hightide embraces REST architecture as much as possible. So, Resources are a fundamental part of the framework. In conjuction with routing they propably are the most significant parts of Hightide.

7.1 Adding validation

7.2 Storing in a database

CHAPTER 8

Processing the requests

CHAPTER 9

Preparing the response

CHAPTER 10

Securing your application

CHAPTER 11

Internationalization

CHAPTER 12

It’s all about testing

CHAPTER 13

Preparing your release

CHAPTER 14

Indices and tables

- *genindex*
- *modindex*
- *search*