Circumflex

Web Framework

Introduction

Circumflex Software Foundation

- Web Framework
- ORM Framework
- Markeven
- Docco
- tools & integration projects

COMMON GOALS

- focus on code quality
- make minimal assumptions
- provide thin abstractions
- make the life easier

Circumflex

Web Framework

Architecture Overview

Underlying Technologies

- HTTP protocol
- Java Servlet API 2.5
- Standard Servlet Containers (Jetty, Tomcat, GlassFish, Resin, IBM WebSphere AS, etc...)
- Scala 2.8.1 (yet)

Model View Controller

The developer's choice:

- FreeMarker
- Scalate
- Scala XML snippets
- Plaintext
- integrate other libraries

- Circumflex ORM
- Squeryl
- NoSQL solutions
- XML solutions
- other tools

Circumflex Web Framework

Circumflex

Web Framework

Application Lifecycle

The Routing Paradigm

User Request

```
GET /hello/world HTTP/1.1
```

Main Request Router

```
class Main extends RequestRouter {

  get("/") = <h1>Welcome!</h1>

  get("/hello") = <h1>Hello world!</h1>

  get("/hello/:name") =
    <h1>Hello, {param("name")}!</h1>

}
```

Handling Complexity

Delegation & Prefixes

```
class Main extends RequestRouter {
  new FooRouter
  new BarRouter
}

class FooRouter extends RequestRouter("/foo") {
  get("/?") = { ... }    // matches `/foo` and `/foo/`
  ...
}

class BarRouter extends RequestRouter("/bar") {
  ...
}
```

Handling Complexity

Subroutes

```
any(“/users/:id/*”) = User.find(param(“id”)) map { user =>
  subroute(“/users/” + user.id) {

    get(“/?” = {...}  // matches `/users/10` or `/users/10/`
    get(“/profile”) = {...}   // matches `/users/10/profile`

    ...
  }
} getOrElse sendError(404)    // if user does not exist
```

Request Dispatching

- Single thread for request processing
  - data sharing using `ThreadLocal` context
  - reliable Java EE hardware scalability
  - transaction-per-request pattern
- Request router instance per request
  - thread safety
  - mutable state
- Exception-driven flow
  - using Scala's `ControlThrowable`

Template Engines Integration

Template Engine

```
def process(template: String, data: Any): String
```

Data is shared using `Context`

```
get("/users/:id") = User.get(param("id")) map { user =>
  'user := user
  ftl("/users/view.ftl")
} getOrElse sendError(404)
```

```
[#ftl]
<h1>Hello, ${user.name}</h1>
```

Circumflex

Web Framework

API Overview

Request API

- accessing request parameters, headers, cookies, etc.
- manipulating with Servlet sessions
- parsing multi-part requests with Apache Commons FileUpload
- accessing low-level `HttpServletRequest`

Response API

- modifying response state: status code, cookies, content type, etc.
- helpers for sending files
- low-level `HttpServletResponse`

Miscellaneous

- Matchers API
- Routing API
- Testing API
- Standalone Server using Embedded Jetty

Circumflex

Web Framework

HOWTO

Before installation

Build automation

maven.apache.org

Java 6 SDK

oracle.com

Revision control

git-scm.com

Development environment

jetbrains.com/idea/

Building Circumflex

- Clone Circumflex repository:

```
cd Projects
git clone git://github.com/inca/circumflex.git
```

- Build with Maven

```
cd circumflex
mvn clean install
```

Generating from Archetype

- Generate new project:

```
cd Projects
mvn archetype:generate
```

- Compiling:

```
cd wiki-slides
mvn compile jetty:run
```

Simple Circumflex Applicatio

localhost:8180

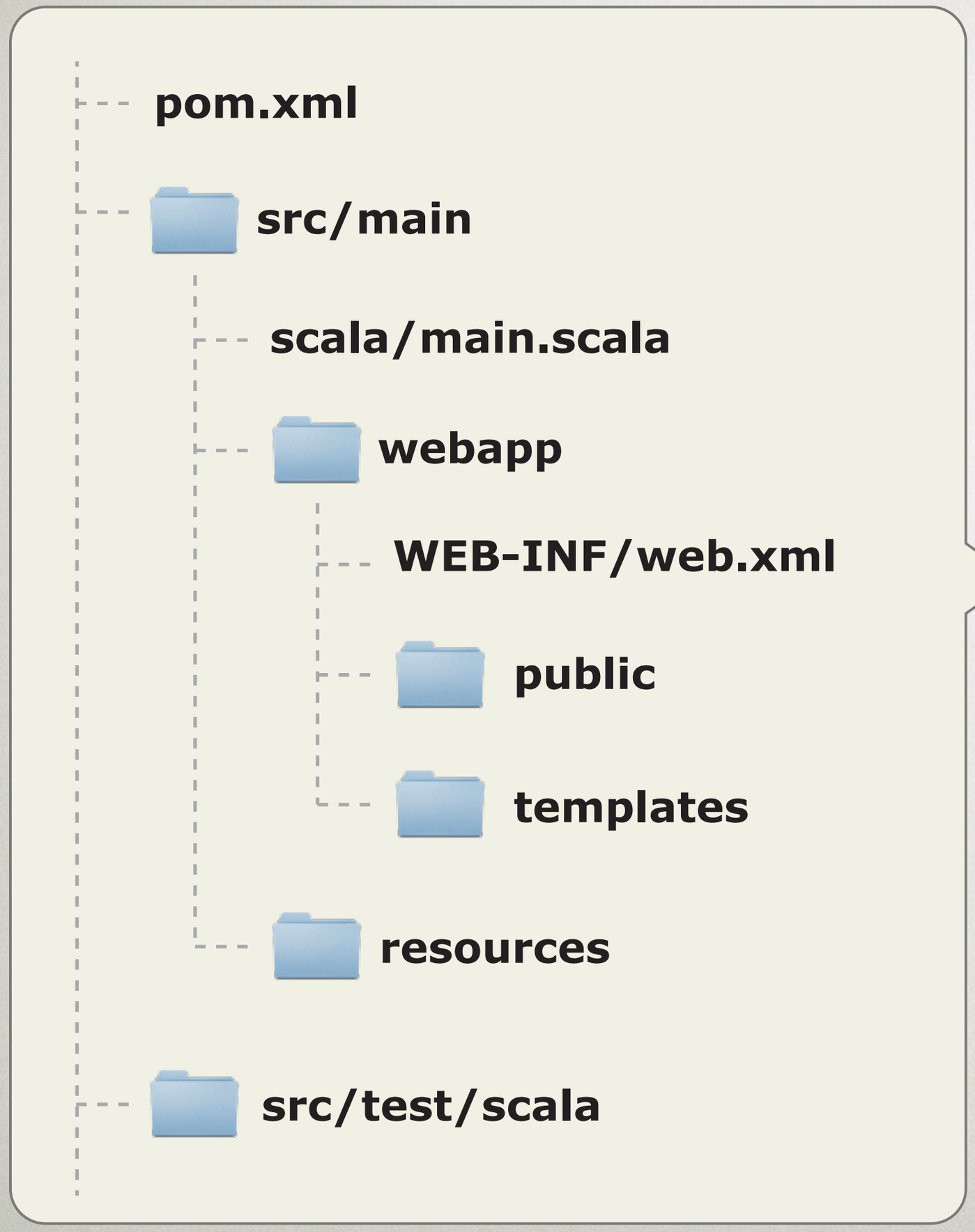
Welcome to Simple Circumflex Web Application!

Please take a moment to observe your project layout.

- **pom.xml** — Maven2 Project Object Model, an XML file that contains information about the project and configuration details used by Maven to build the project. It contains sensible default values for most projects. Basically you configure your project's dependencies, build names, source and target directories and build plugins using this configuration file. For more information, proceed to Introduction to the POM.
- **src/main** — main source directory of your project:
  - **scala/main.scala** — your application's entry point called *Request Router*, it handles all incoming HTTP requests, matching them against *routes*: the first route that matches request is executed;
  - **webapp** — your Web application context root:
    - **WEB-INF/web.xml** — the Java EE *deployment descriptor*, an XML configuration file that specifies, how your application will be deployed into *Servlet Container*. By default it just maps all URLs to *Circumflex Filter*.
    - **public** — a directory for static resources, e.g. stylesheets, scripts, images, etc.; resources under this directory are served directly by the container and are not processed by *Circumflex Filter*.
    - **templates** — a default location for FreeMarker templates; they are resolved relatively to this location by *Circumflex FreeMarker helper*.
  - **resources/Messages.properties** — various messages for your application can be defined there for Internationalization purposes;
  - **resources/cx.properties** — Circumflex configuration parameters are specified here, by default it only contains the *cx.router* parameter, which points to your application Request Router class;
- **src/test/scala** — your tests source directory. It contains a simple spec out-of-box. You are free to add more sophisticated tests for your application.

Please visit Circumflex website for further information.

We hope you'll get a lot of fun developing with Circumflex! Good luck!



Project Layout

Sample Application

http://github.com/lirio/wiki-slides

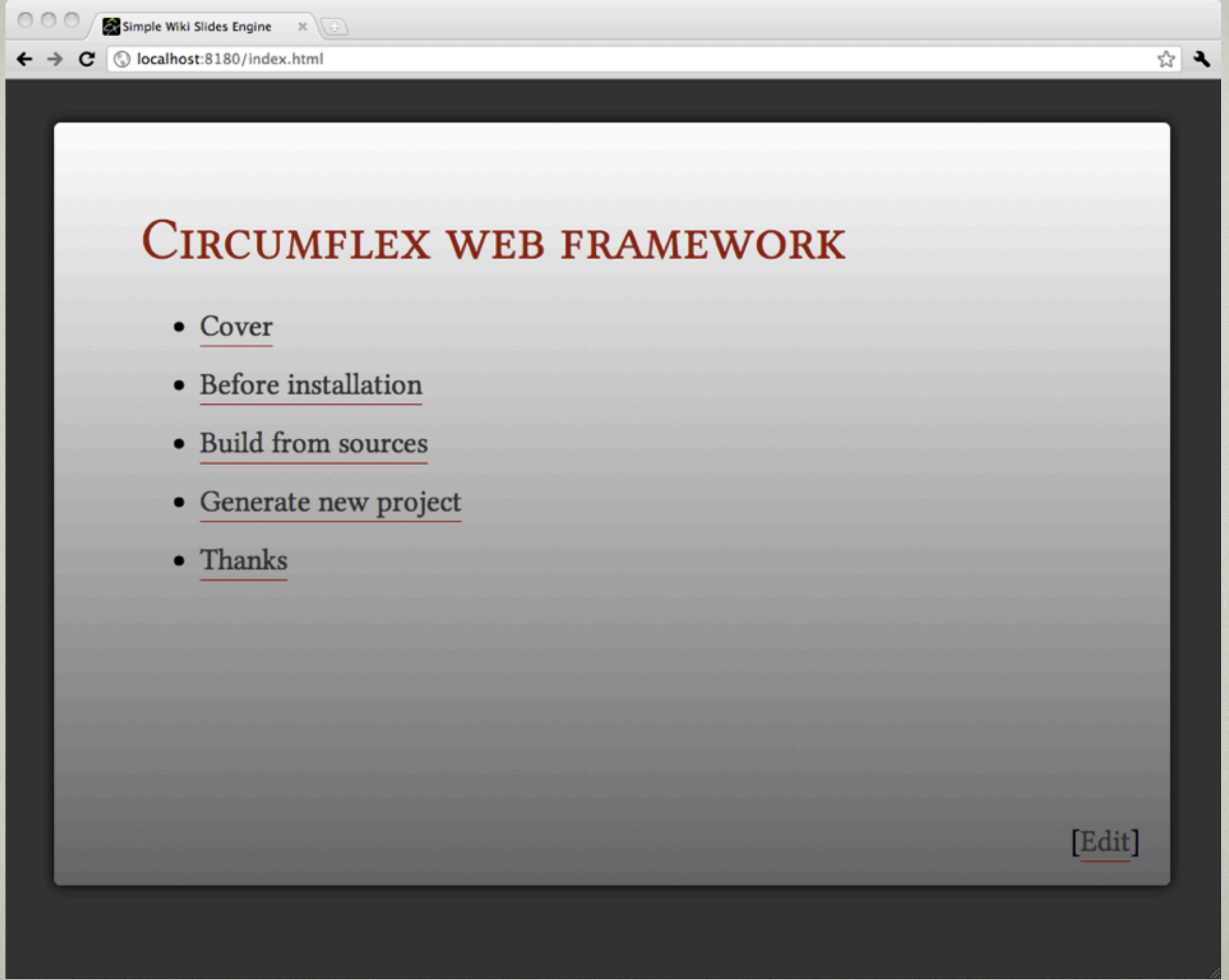
Simple Wiki Slides Engine

localhost:8180

Circumflex

Web Framework

Introduction


Simple Wiki Slides Engine
localhost:8180/index.html
Circumflex web framework
Cover
Before installation
Build from sources
Generate new project
Thanks
[Edit]

Simple Wiki Slides Engine

localhost:8180/index.html.e

INDEX

```
# Circumflex web framework

* [Cover](cover.html)                    {.slides-refs}
* [Before installation](requirements.html)
* [Build from sources](build.html)
* [Generate new project](generate.html)
* [Thanks](thanks.html)
```

Save

References

- Circumflex Main Site
  http://circumflex.ru

- Circumflex Sources
  http://github.com/inca/circumflex

- FreeMarker Template Engine
  http://freemarker.org

Thanks!