play framework reference Documentation

1.0

nummy

2016 10 29

Contents

:

CHAPTER 1

ActionsControllersResults

1.1 Action

Play `Action Action (play.api.mvc.Request => play.api.mvc.Result)`

```
def echo = Action { request =>
  Ok("Got request [" + request + "]")
}
```

1.2 Action

```
Action {
  Ok("Hello world")
}
```

```
Action {implicit request =>
  Ok("Got request [" + request + "]")
}
```

Bodyparse

```
Action(parse.json) { implicit request =>
  Ok("Got request [" + request + "]")
}
```

1.3 ControllersAction

Controllers Action

```
package controllers

import play.api.mvc._

class Application extends Controller {

  def index = Action {
    Ok("It works!")
  }
```

```
}
```

1.4

play.api.mvc.Result

```
import play.api.http.HttpEntity

def index = Action {
  Result(
    header = ResponseHeader(200, Map.empty),
    body = HttpEntity.Strict(ByteString("Hello world!"), Some("text/plain"))
  )
}
```

play Ok() :

```
def index = Action {
  Ok("Hello world!")
}
```

```
val ok = Ok("Hello world!")
val notFound = NotFound
val pageNotFound = NotFound(<h1>Page not found</h1>)
val badRequest = BadRequest(views.html.form(formWithErrors))
val oops = InternalServerError("Oops")
val anyStatus = Status(488)("Strange response type")
```

1.5

```
def index = Action {
  Redirect("/user/home")
}
```

1.6 TODO

```
def index(name:String) = TODO
```

CHAPTER 2

HTTP

2.1 HTTP

HTTPAction

HTTPMVC

-
-

`conf/routes` Play

2.2

Play

-
-

`build.sbt`

```
routesGenerator := StaticRoutesGenerator
```

2.3

`conf/routes` URLAction

```
GET    /clients/:id          controllers.Clients.show(id: Long)
```

URLAction

#

```
# Display a client.
GET    /clients/:id          controllers.Clients.show(id: Long)
```

->

```
->      /api                        api.MyRouter
```

2.4 HTTP

PlayHTTPGET,POST,PUT, DELETE, HEAD

2.5 URL

URL

2.5.1

GET /clients/all

```
GET    /clients/all          controllers.Clients.list()
```

2.5.2

client id

```
GET    /clients/:id          controllers.Clients.show(id: Long)
```

[^/]+

/ URL *id .*

```
GET    /files/*name          controllers.Application.download(name)
```

GET /files/images/logo.png name images/logo.png

PlayURL $id<regex>

```
GET    /items/$id<[0-9]+>    controllers.Items.show(id: Long)
```

2.6 Action

ActionAction

GET / controllers.Application.homePage()

URI

```
# Extract the page parameter from the path.
GET    /:page                controllers.Application.show(page)

# Extract the page parameter from the query string.
GET    /                     controllers.Application.show(page)
```

```
def show(page: String) = Action {
  loadContentFromDatabase(page).map { htmlContent =>
    Ok(htmlContent).as("text/html")
  }.getOrElse(NotFound)
}
```

2.6.1

String Scala

```
GET    /clients/:id          controllers.Clients.show(id: Long)
```

show

```
def show(id: Long) = Action {
  Client.findById(id).map { client =>
    Ok(views.html.Clients.display(client))
  }.getOrElse(NotFound)
}
```

2.6.2

2.6.3

```
# Pagination links, like /clients?page=3
GET    /clients              controllers.Clients.list(page: Int ?= 1)
```

2.6.4

```
# The version parameter is optional. E.g. /api/list-all?version=3.0
GET    /api/list-all         controllers.Api.list(version: Option[String])
```

2.7

2.8

URL `controller` playroutes `play.api.mvc.Call`

`play.api.mvc.Call` HTTPURI

```
package controllers

import play.api._
import play.api.mvc._

class Application extends Controller {

  def hello(name: String) = Action {
    Ok("Hello " + name + "!")
  }

}
```

```
# Hello action
GET    /hello/:name          controllers.Application.hello(name)
```

`hello` URL

```
// Redirect to /hello/Bob
def helloBob = Action {
  Redirect(routes.Application.hello("Bob"))
}
```

2.9

Play

```
# Redirects to https://www.playframework.com/ with 303 See Other
GET   /about      controllers.Default.redirect(to = "https://www.playframework.com/")

# Responds with 404 Not Found
GET   /orders     controllers.Default.notFound

# Responds with 500 Internal Server Error
GET   /clients    controllers.Default.error

# Responds with 501 Not Implemented
GET   /posts      controllers.Default.todo
```

3.1 Content-Type

Play

```
val textResult = Ok("Hello World!")
```

Content-Typetext/plain

```
val xmlResult = Ok(<message>Hello World!</message>)
```

Content-Typeapplication/xml

play.api.http.ContentType

```
val htmlResult = Ok(<h1>Hello World!</h1>).as("text/html")
```

```
val htmlResult2 = Ok(<h1>Hello World!</h1>).as(HTML)
```

3.2 HTTP headers

HTTP

```
val result = Ok("Hello World!").withHeaders(
        CACHE_CONTROL -> "max-age=3600",
        ETAG -> "xx")
```

3.3 Cookie

Cookie

```
val result = Ok("Hello world").withCookies(
        Cookie("theme", "blue"))
```

Cookie

```
val result2 = result.discardingCookies(DiscardingCookie("theme"))
```

Cookie

```
val result3 = result.withCookies(Cookie("theme", "blue")).discardingCookies(DiscardingCookie("skin"))
```

3.4

PlayUTF-8

```
import play.api.mvc.Codec

class Application extends Controller {

        implicit val myCustomCharset = Codec.javaSupported("iso-8859-1")

        def index = Action {
        Ok(<h1>Hello World!</h1>).as(HTML)
        }

}
```

CHAPTER 4

SessionFlash

HTTP `Session` `Flash`

- `Session`
- `Flash`

Play `session` `flash` 4kb `cookie` `PLAY_SESSION` `session.cookieName`

4.1 session

```
Ok("Welcome!").withSession(
  "connected" -> "user@gmail.com")
```

`session`

```
Ok("Hello World!").withSession(
  request.session + ("saidHello" -> "yes"))
```

`session`

```
Ok("Theme reset!").withSession(
  request.session - "theme")
```

4.2 session

HTTP `session`

```
def index = Action { request =>
  request.session.get("connected").map { user =>
    Ok("Hello " + user)
  }.getOrElse {
    Unauthorized("Oops, you are not connected")
  }
```

4.3

```
Ok("Bye").withNewSession
```

CHAPTER 5

body parser

HTTP

`RequestHeader BodyParser`

Play `InputStream` Play `Akka Streams`

5.1

webPlay `Content-Type`

CHAPTER 6

action

6.1 action

action ActionBuilder action ActionBuilder

action ActionBuilder invokeBlock action

```
import play.api.mvc._

object LoggingAction extends ActionBuilder[Request] {
  def invokeBlock[A](request: Request[A], block: (Request[A]) => Future[Result]) = {
    Logger.info("Calling action")
    block(request)
  }
}
```

LogginAction

```
def index = LoggingAction {
  Ok("Hello World")
}
```

6.2 action

Action

7.1

HTTP `Accept`

```
val list = Action { implicit request =>
  val items = Item.findAll
  render {
    case Accepts.Html() => Ok(views.html.list(items))
    case Accepts.Json() => Ok(Json.toJson(items))
  }
}
```

`Accepts.Html()` `Accepts.Json()`

7.2

Play `Accepts` MIME

- Xml
- Html
- Json
- JavaScript

`MIME` ‘‘play.api.mvc.Accepting‘‘

HTTP:

-
-

Play

Play `Action` Play

Play `HttpErrorHanlder`

- onClientError
- onServerError

8.1

`ErrorHandler HttpErrorHandler`

```
import play.api.http.HttpErrorHandler
import play.api.mvc._
import play.api.mvc.Results._
import scala.concurrent._
import javax.inject.Singleton;

@Singleton
class ErrorHandler extends HttpErrorHandler {

  def onClientError(request: RequestHeader, statusCode: Int, message: String) = {
    Future.successful(
      Status(statusCode)("A client error occurred: " + message)
    )
  }

  def onServerError(request: RequestHeader, exception: Throwable) = {
    Future.successful(
      InternalServerError("A server error occurred: " + exception.getMessage)
    )
  }
}
```

`application.conf`

```
play.http.errorHandler = "com.example.ErrorHandler"
```

8.2 Error Handler

Play ··DefaultHttpErrorHandler··

```
import javax.inject._

import play.api.http.DefaultHttpErrorHandler
import play.api._
import play.api.mvc._
import play.api.mvc.Results._
import play.api.routing.Router
import scala.concurrent._

@Singleton
class ErrorHandler @Inject() (
    env: Environment,
    config: Configuration,
    sourceMapper: OptionalSourceMapper,
    router: Provider[Router]
  ) extends DefaultHttpErrorHandler(env, config, sourceMapper, router) {

  override def onProdServerError(request: RequestHeader, exception: UsefulException) = {
    Future.successful(
      InternalServerError("A server error occurred: " + exception.getMessage)
    )
  }

  override def onForbidden(request: RequestHeader, message: String) = {
    Future.successful(
      Forbidden("You're not allowed to access this resource.")
    )
  }
}
```

CHAPTER 9

HTTP

9.1

PlayHTTP

Play `controller` Play `action`

`future`

`Future[Result] Result Future[Result] Result`

`Future[Result] Future Result`

```
import play.api.libs.concurrent.Execution.Implicits.defaultContext

val futurePIValue: Future[Double] = computePIAsynchronously()
val futureResult: Future[Result] = futurePIValue.map { pi =>
  Ok("PI value computed: " + pi)
}
```

9.2

`apply action Action.async`

```
import play.api.libs.concurrent.Execution.Implicits.defaultContext

def index = Action.async {
  val futureInt = scala.concurrent.Future { intensiveComputation() }
  futureInt.map(i => Ok("Got result: " + i))
}
```

`Actions`

9.3

`promise timeout`

```
import play.api.libs.concurrent.Execution.Implicits.defaultContext
import scala.concurrent.duration._
```

```
def index = Action.async {
  val futureInt = scala.concurrent.Future { intensiveComputation() }
  val timeoutFuture = play.api.libs.concurrent.Promise.timeout("Oops", 1.second)
  Future.firstCompletedOf(Seq(futureInt, timeoutFuture)).map {
    case i: Int => Ok("Got result: " + i)
    case t: String => InternalServerError(t)
  }
}
```

9.4

9.5

Play

```
def index = Action {
 Ok.sendFile(new java.io.File("/tmp/fileToServe.pdf"))
}
```

Play Conten-Type Content-Disposition

```
def index = Action {
  Ok.sendFile(
    content = new java.io.File("/tmp/fileToServe.pdf"),
    fileName = _ => "termsOfService.pdf"
 )
}
```

```
def index = Action {
  Ok.sendFile(
    content = new java.io.File("/tmp/fileToServe.pdf"),
    inline = true
  )
}
```

9.6

```
def index = Action {
  val CHUNK_SIZE = 100
  val data = getDataStream
  val dataContent: Source[ByteString, _] = StreamConverters.fromInputStream(data, CHUNK_SIZE)

  Ok.chunked(dataContent)
}
```

Source

```
def index = Action {
  val source = Source.apply(List("kiki", "foo", "bar"))
  Ok.chunked(source)
}
```

CHAPTER 10

PlayTwirlscala

-
-
- scala
-

10.1

10.1.1

Play ScalascalaHTMLXMLCSV

Scala `views/Application/index.scala.html views.html.Application.index apply()`

```
@(customer: Customer, orders: List[Order])

<h1>Welcome @customer.name!</h1>

<ul>
@for(order <- orders) {
  <li>@order.title</li>
}
</ul>
```

Scala

```
val content = views.html.Application.index(c, o)
```

10.1.2 @

Scala `@` scalascalaScala

```
Hello @customer.name!
       ^^^^^^^^^^^^^
       Dynamic code
```

scala

scala

```
Hello @{val name = customer.firstName + customer.lastName; name}!
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                              Dynamic Code
```

@ @@

```
My email is bob@@example.com
```

10.1.3

scala

```
@(customer: Customer, orders: List[Order])
```

```
@(title: String = "Home")
```

```
@(title: String)(body: Html)
```

10.1.4

scala for

```
<ul>
@for(p <- products) {
  <li>@p.name ($@p.price)</li>
}
</ul>
```

{ for

10.1.5

scala

```
@if(items.isEmpty) {
  <h1>Nothing to display</h1>
} else {
  <h1>@items.size items!</h1>
}
```

10.1.6

```
@display(product: Product) = {
  @product.name ($@product.price)
}

<ul>
@for(product <- products) {
  @display(product)
```

```
}
</ul>
```

scala

```
@title(text: String) = @{
  text.split(' ').map(_.capitalize).mkString(" ")
}

<h1>@title("hello world")</h1>
```

implicit implicit

```
@implicitFieldConstructor = @{ MyFieldConstructor() }
```

10.1.7

defining

```
@defining(user.firstName + " " + user.lastName) { fullName =>
  <div>Hello @fullName</div>
}
```

10.1.8

```
@(customer: Customer, orders: List[Order])
@import utils._
```

root

```
@import _root_.company.product.core._
```

build.sbt

```
TwirlKeys.templateImports += "org.abc.backend._"
```

10.1.9

@**@:

```
@*********************
  * This is a comment *
*********************@
```

ScalaAPI

```
@************************************
* Home page.                        *
*                                   *
* @param msg The message to display *
************************************@
@(msg: String)

<h1>@msg</h1>
```

10.1.10

```
<p>
  @Html(article.content)
</p>
```

10.2

Play

10.2.1 Layout

views/main.scala.html

title HTML views/Application/index.scala.html

```
@main(title = "Home") {

  <h1>Home page</h1>

}
```

```
@(title: String)(sidebar: Html)(content: Html)
<!DOCTYPE html>
<html>
  <head>
    <title>@title</title>
  </head>
  <body>
    <section class="sidebar">@sidebar</section>
    <section class="content">@content</section>
  </body>
</html>
```

```
@sidebar = {
  <h1>Sidebar</h1>
}

@main("Home")(sidebar) {
  <h1>Home page</h1>

}
```

CHAPTER 12

JSON

12.1 JSON

12.2 JSONHTTP

12.3 JSON

12.4 JSON

12.5 JSON

CHAPTER 13

XML

14.1

HTML

```
public Result upload() {
    MultipartFormData<File> body = request().body().asMultipartFormData();
    FilePart<File> picture = body.getFile("picture");
    if (picture != null) {
        String fileName = picture.getFilename();
        String contentType = picture.getContentType();
        File file = picture.getFile();
        return ok("File uploaded");
    } else {
        flash("error", "Missing file");
        return badRequest();
    }
}
```

14.2

JSON

```
public Result upload() {
    File file = request().body().asRaw().asFile();
    return ok("File uploaded");
}
```

CHAPTER 16

Play EHCache API

16.1

build.sbt

```
libraryDependencies ++= Seq(
  cache
)
```

16.2 API

API CacheApi

```
import play.api.cache._
import play.api.mvc._
import javax.inject.Inject

class Application @Inject() (cache: CacheApi) extends Controller {

}
```

16.2.1

```
cache.set("item.key", connectedUser)
```

.. code-block:: scala

    import scala.concurrent.duration._

    cache.set("item.key", connectedUser, 5.minutes)

16.2.2

```
val user: User = cache.getOrElse[User]("item.key") {
  User.findById(connectedUser)
}
```

16.2.3

```
cache.remove("item.key");
```

16.3

play application.conf

```
play.cache.bindCaches = ["db-cache", "user-cache", "session-cache"]
```

NamedCache

```
import play.api.cache._
import play.api.mvc._
import javax.inject.Inject

class Application @Inject()(
    @NamedCache("session-cache") sessionCache: CacheApi
) extends Controller {

}
```

16.4 HTTP

PlayHTTP Cached

```
import play.api.cache.Cached
import javax.inject.Inject

class Application @Inject() (cached: Cached) extends Controller {

}
```

```
def index = cached("homePage") {
  Action {
    Ok("Hello world")
  }
}
```

key

‘‘‘

16.5

200

```
def get(index: Int) = cached.status(_ => "/resource/"+ index, 200) {
  Action {
    if (index > 0) {
      Ok(Json.obj("id" -> index))
    } else {
      NotFound
```

```
    }
  }
}
```

404:

```
def get(index: Int) = {
  val caching = cached
    .status(_ => "/resource/"+ index, 200)
   .includeStatus(404, 600)

 caching {
    Action {
      if (index % 2 == 1) {
       Ok(Json.obj("id" -> index))
      } else {
       NotFound
      }
    }
 }
}
```

CHAPTER 17

web

CHAPTER 18

Akka

Akka

18.1 actor

Akka `actor` `actor`

Play `actor` `play`

18.1.1 actors

Akka `actor`

```
import akka.actor._

object HelloActor {
 def props = Props[HelloActor]

 case class SayHello(name: String)
}

class HelloActor extends Actor {
  import HelloActor._

  def receive = {
    case SayHello(name: String) =>
      sender() ! "Hello, " + name
  }
}
```

`actor` Akka

-
- props

18.2 actors

actor `ActorSystem`

```
import play.api.mvc._
import akka.actor._
import javax.inject._

import actors.HelloActor

@Singleton
class Application @Inject() (system: ActorSystem) extends Controller {

  val helloActor = system.actorOf(HelloActor.props, "hello-actor")

 //...
}
```

actorOf actor Controller actor actor

18.3

actor actor

HTTP Future

```
import play.api.libs.concurrent.Execution.Implicits.defaultContext
import scala.concurrent.duration._
import akka.pattern.ask
implicit val timeout: Timeout = 5.seconds

def sayHello(name: String) = Action.async {
  (helloActor ? SayHello(name)).mapTo[String].map { message =>
    Ok(message)
  }
}
```

:

-
- Future[Result]
- timeout

18.4 actors

CHAPTER 20

CHAPTER 21