Scala Exercises

WHAT IS IT?

A BROWSER TOOL TO LEARN SCALA AND ITS COOLEST LIBS

A BIT OF HISTORY

- @RAFAPARADELA JS POC FOR THE @47DEG LABS
  - V1 WENT LIVE ON MARCH 2ND. 2015
- FOCUSED ON PORTING SCALA KOANS TO THE WEB
  - SOCIAL COMPONENT

V1 LIMITATIONS

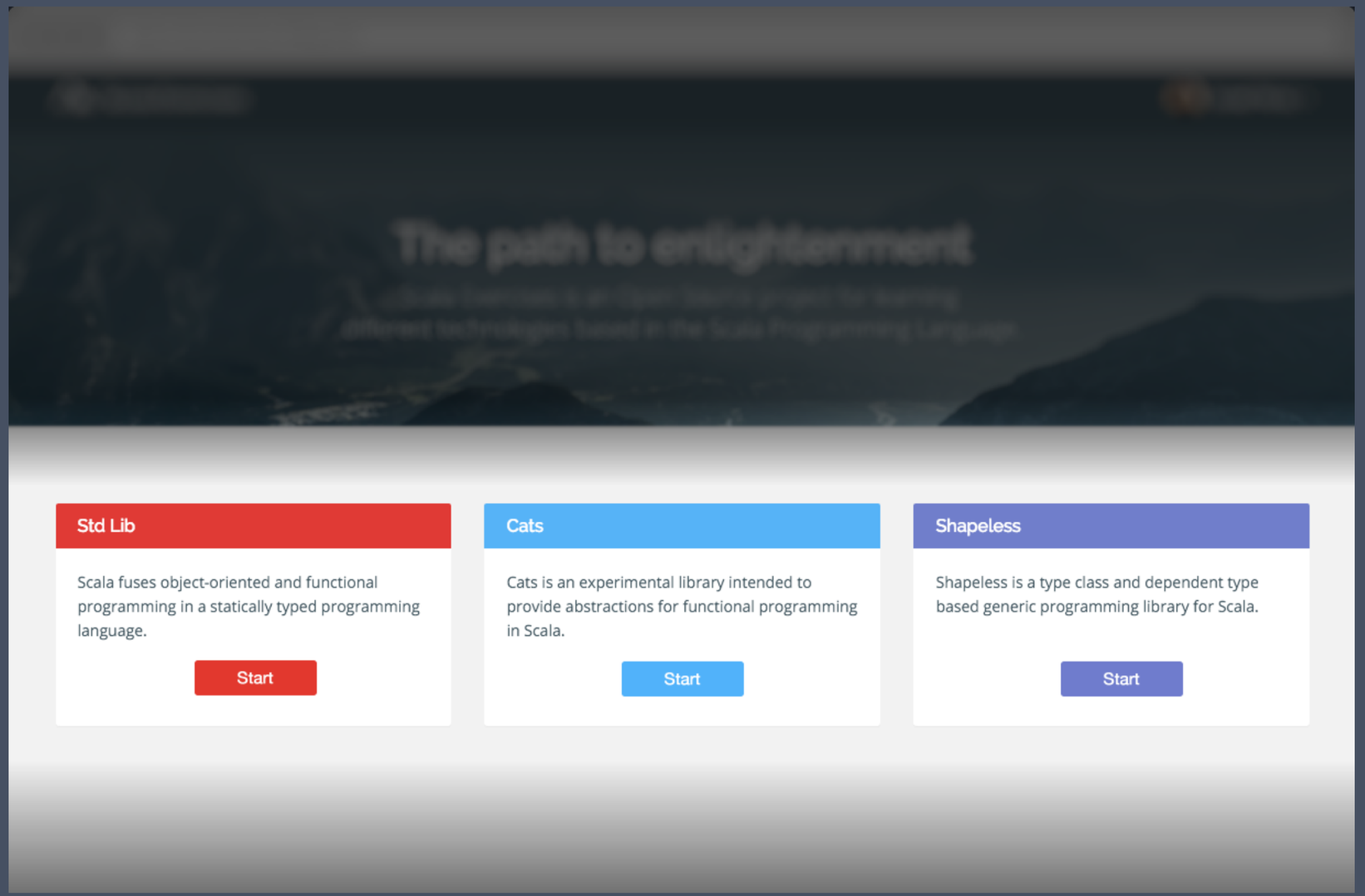
- SERVERLESS
- "EVALUATION" CONSTRAINED TO EXACT STRING MATCHES
- CAN'T TRACK USER PROGRESS ACROSS DEVICES
- CAN'T PARSE OR EVALUATE MORE COMPLEX EXERCISES

V1 STATS

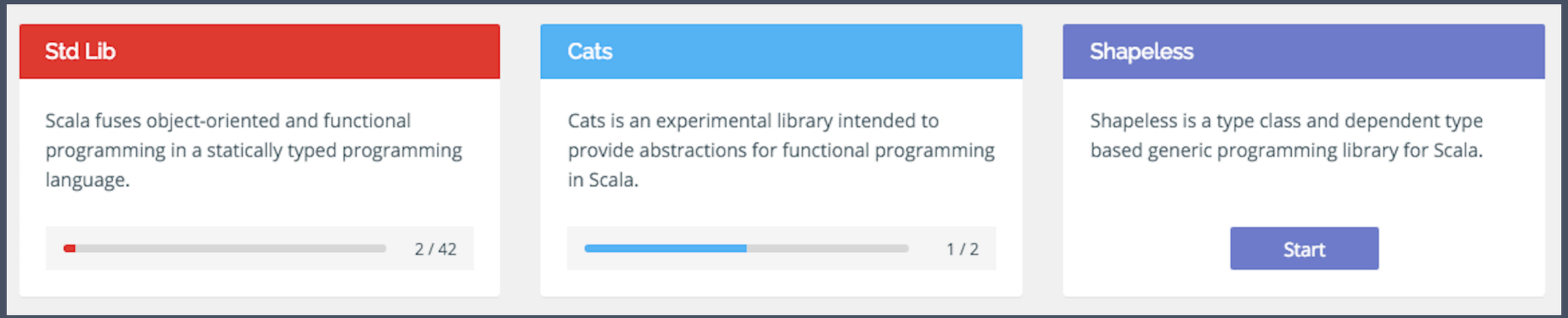
- 63K SESSIONS
- 40% RETURNING
- 180 PRS
- 50 CONTRIBUTORS
- 1 SECTION COMPLETED PER VISIT

WHAT'S NEW IN V2?

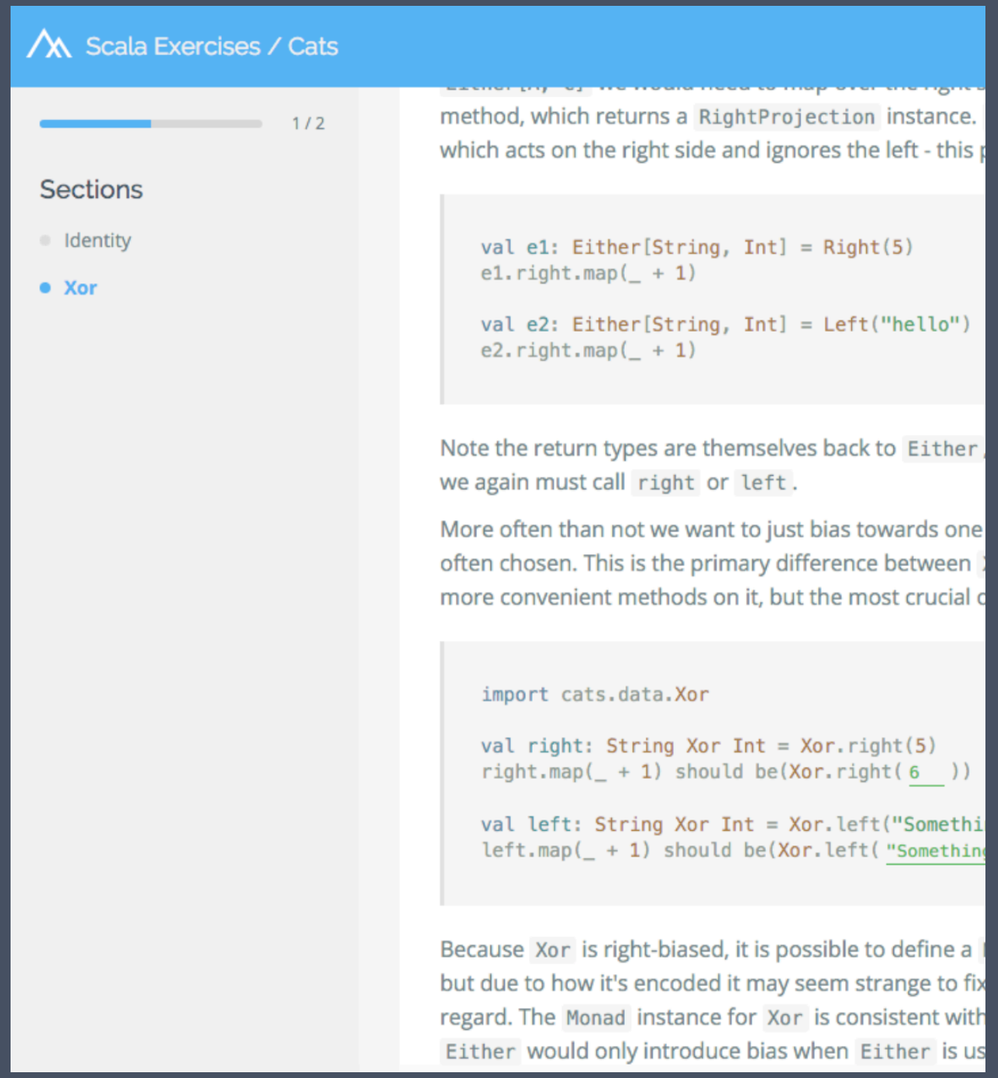
LIBS AT A GLANCE



OVERALL PROGRESS



LIB PROGRESS



WRITE EXERCISES WITH CODE!

```scala
package catslib

import cats._
import org.scalatest._

/** identity
  *
  * The identity monad can be seen as the ambient monad that encodes the
  * effect of having no effect. It is ambient in the sense that plain pure
  * values are values of `Id`.
  *
  * It is encoded as:
  *
  * {{{
  * type Id[A] = A
  * }}}
  *
  * That is to say that the type Id[A] is just a synonym for A.  We can
  * freely treat values of type `A` as values of type `Id[A]`, and
  * vice-versa.
  *
  * {{{
  * import cats._
  *
  * val x: Id[Int] = 1
  * val y: Int = x
  * }}}
  *
  */
object IdentitySection extends FlatSpec with Matchers with exercise.Section {

  /** We can freely compare values of `Id[T]` with unadorned
    * values of type `T`.
    */
  def identityType(res0: Int) {
    val anId: Id[Int] = 42
    anId should be (res0)
  }

```

http://scala-exercises.47deg.com/

Scala Exercises / Cats

Identity

The identity monad can be seen as the ambient monad that encodes the effect of having no effect. It is ambient in the sense that plain pure values are values of `Id`.

It is encoded as:

```
type Id[A] = A
```

That is to say that the type Id[A] is just a synonym for A. We can freely treat values of type `A` as values of type `Id[A]`, and vice-versa.

```
import cats._

val x: Id[Int] = 1
val y: Int = x
```

We can freely compare values of `Id[T]` with unadorned values of type `T`.

```
val anId: Id[Int] = 42
anId should be ( _ )
```

▶ Run

WRITE EXERCISES WITH CODE!

```scala
package catslib

import cats._
import org.scalatest._

/** identity
  *
  * The identity monad can be seen as the ambient monad that encodes the
  * effect of having no effect. It is ambient in the sense that plain pure
  * values are values of `Id`.
  *
  * It is encoded as:
  *
  * {{{
  * type Id[A] = A
  * }}}
  *
  * That is to say that the type Id[A] is just a synonym for A.  We can
  * freely treat values of type `A` as values of type `Id[A]`, and
  * vice-versa.
  *
  * {{{
  * import cats._
  *
  * val x: Id[Int] = 1
  * val y: Int = x
  * }}}
  *
  */
object IdentitySection extends FlatSpec with Matchers with exercise.Section {

  /** We can freely compare values of `Id[T]` with unadorned
    * values of type `T`.
    */
  def identityType(res0: Int) {
    val anId: Id[Int] = 42
    anId should be (res0)
  }
```

http://scala-exercises.47deg.com/

Scala Exercises / Cats

Identity

The identity monad can be seen as the ambient monad that encodes the effect of having no effect. It is ambient in the sense that plain pure values are values of `Id`.

It is encoded as:

```
type Id[A] = A
```

That is to say that the type Id[A] is just a synonym for A. We can freely treat values of type `A` as values of type `Id[A]`, and vice-versa.

```
import cats._

val x: Id[Int] = 1
val y: Int = x
```

We can freely compare values of `Id[T]` with unadorned values of type `T`.

```
val anId: Id[Int] = 42
anId should be (  )
```

▶ Run

WRITE EXERCISES WITH CODE!

```scala
package catslib

import cats._
import org.scalatest._

/** identity
  *
  * The identity monad can be seen as the ambient monad that encodes the
  * effect of having no effect. It is ambient in the sense that plain pure
  * values are values of `Id`.
  *
  * It is encoded as:
  *
  * {{{
  * type Id[A] = A
  * }}}
  *
  * That is to say that the type Id[A] is just a synonym for A.  We can
  * freely treat values of type `A` as values of type `Id[A]`, and
  * vice-versa.
  *
  * {{{
  * import cats._
  *
  * val x: Id[Int] = 1
  * val y: Int = x
  * }}}
  *
  */
object IdentitySection extends FlatSpec with Matchers with exercise.Section {

  /** We can freely compare values of `Id[T]` with unadorned
    * values of type `T`.
    */
  def identityType(res0: Int) {
    val anId: Id[Int] = 42
    anId should be (res0)
  }
```

http://scala-exercises.47deg.com/

Scala Exercises / Cats

Identity

The identity monad can be seen as the ambient monad that encodes the effect of having no effect. It is ambient in the sense that plain pure values are values of `Id`.

It is encoded as:

```
type Id[A] = A
```

That is to say that the type Id[A] is just a synonym for A. We can freely treat values of type `A` as values of type `Id[A]`, and vice-versa.

```
import cats._

val x: Id[Int] = 1
val y: Int = x
```

We can freely compare values of `Id[T]` with unadorned values of type `T`.

```
val anId: Id[Int] = 42
anId should be (  )
```

▶ Run

WRITE EXERCISES WITH CODE!

```scala
package catslib

import cats._
import org.scalatest._

/** identity
  *
  * The identity monad can be seen as the ambient monad that encodes the
  * effect of having no effect. It is ambient in the sense that plain pure
  * values are values of `Id`.
  *
  * It is encoded as:
  *
  * {{{
  * type Id[A] = A
  * }}}
  *
  * That is to say that the type Id[A] is just a synonym for A.  We can
  * freely treat values of type `A` as values of type `Id[A]`, and
  * vice-versa.
  *
  * {{{
  * import cats._
  *
  * val x: Id[Int] = 1
  * val y: Int = x
  * }}}
  *
  */
object IdentitySection extends FlatSpec with Matchers with exercise.Section {

  /** We can freely compare values of `Id[T]` with unadorned
    * values of type `T`.
    */
  def identityType(res0: Int) {
    val anId: Id[Int] = 42
    anId should be (res0)
  }
```

http://scala-exercises.47deg.com/

Scala Exercises / Cats

Identity

The identity monad can be seen as the ambient monad that encodes the effect of having no effect. It is ambient in the sense that plain pure values are values of `Id`.

It is encoded as:

```
type Id[A] = A
```

That is to say that the type Id[A] is just a synonym for A. We can freely treat values of type `A` as values of type `Id[A]`, and vice-versa.

```
import cats._

val x: Id[Int] = 1
val y: Int = x
```

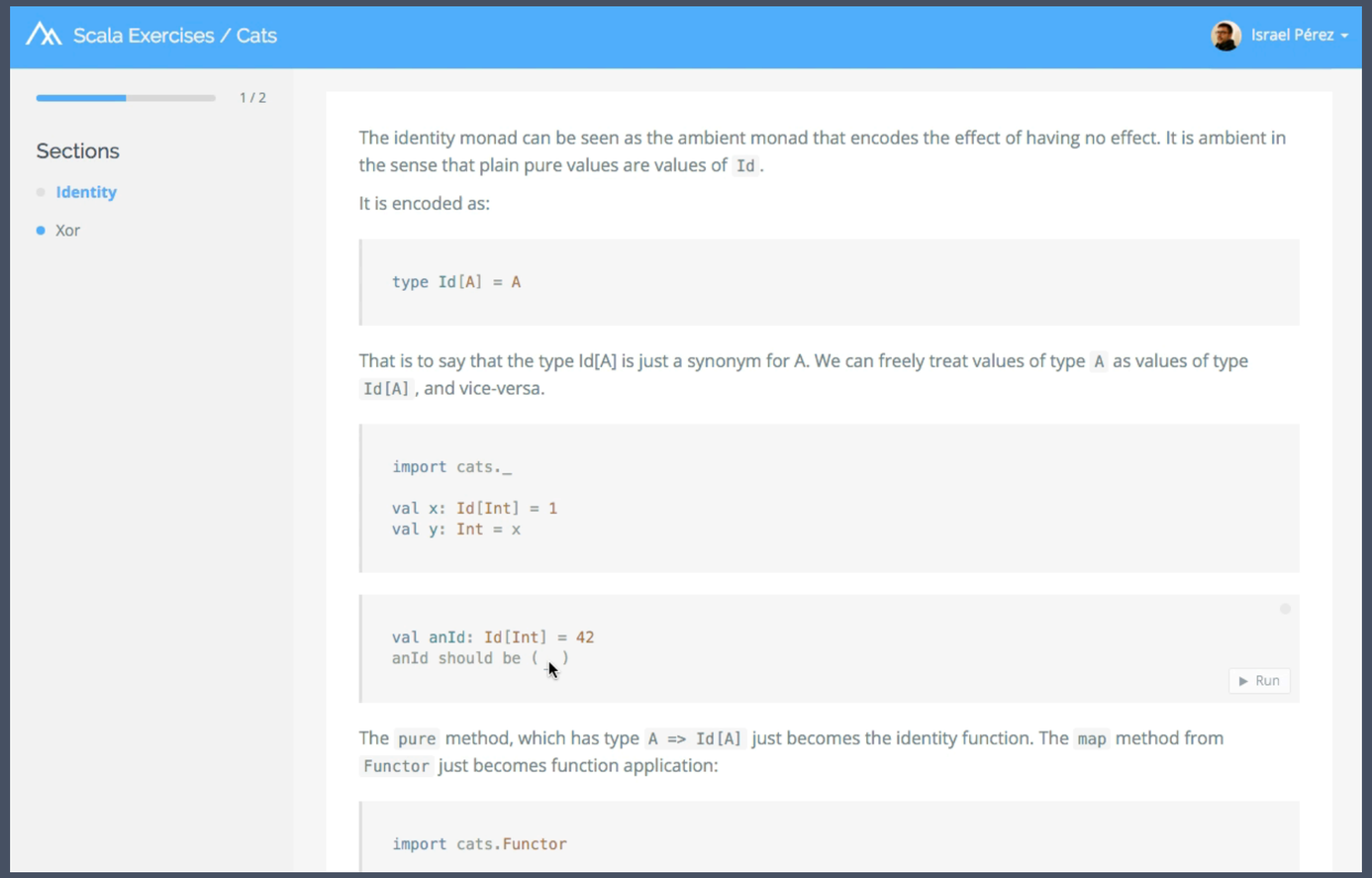
We can freely compare values of `Id[T]` with unadorned values of type `T`.

```
val anId: Id[Int] = 42
anId should be ( __ )
```

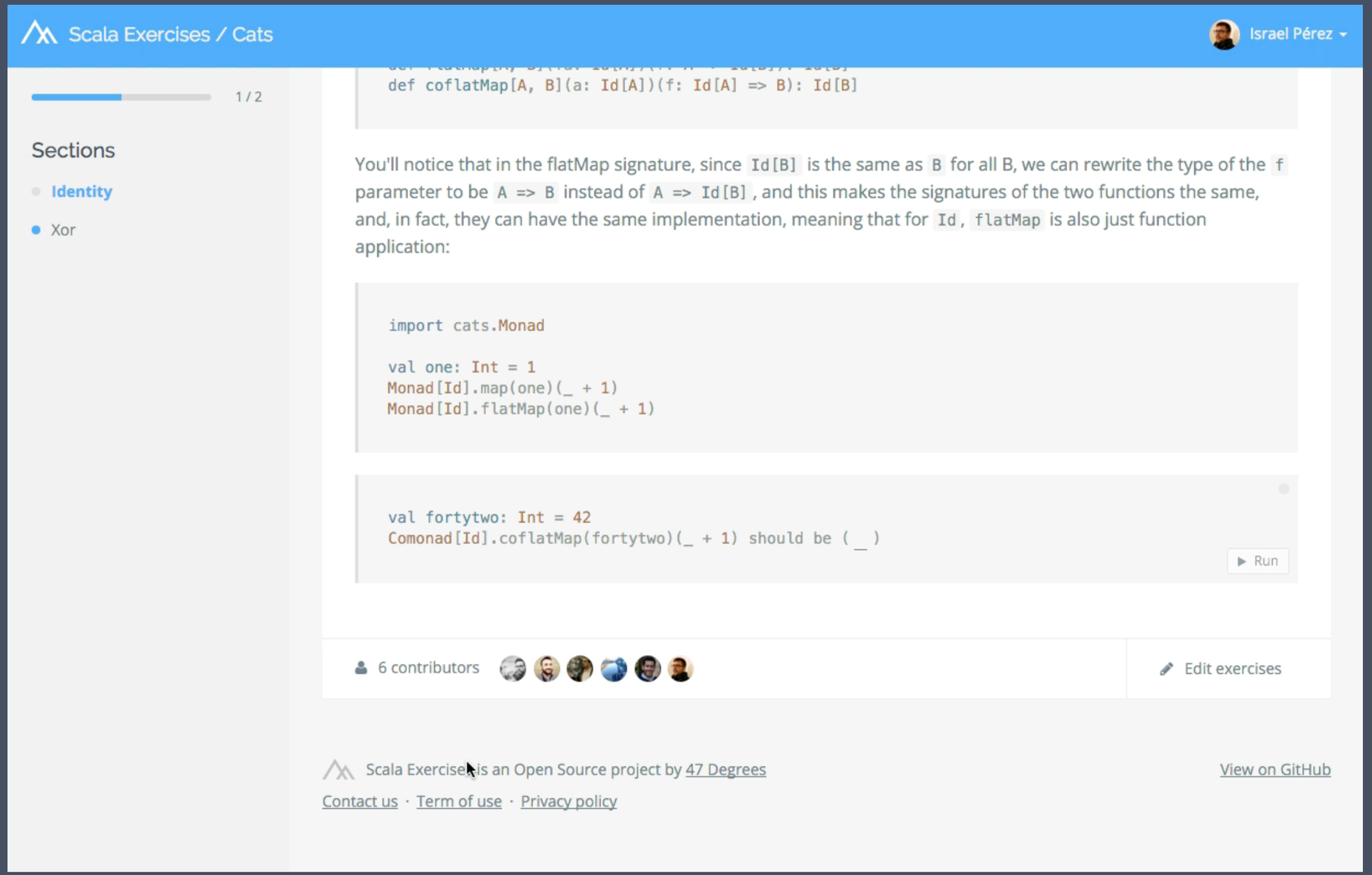
▶ Run

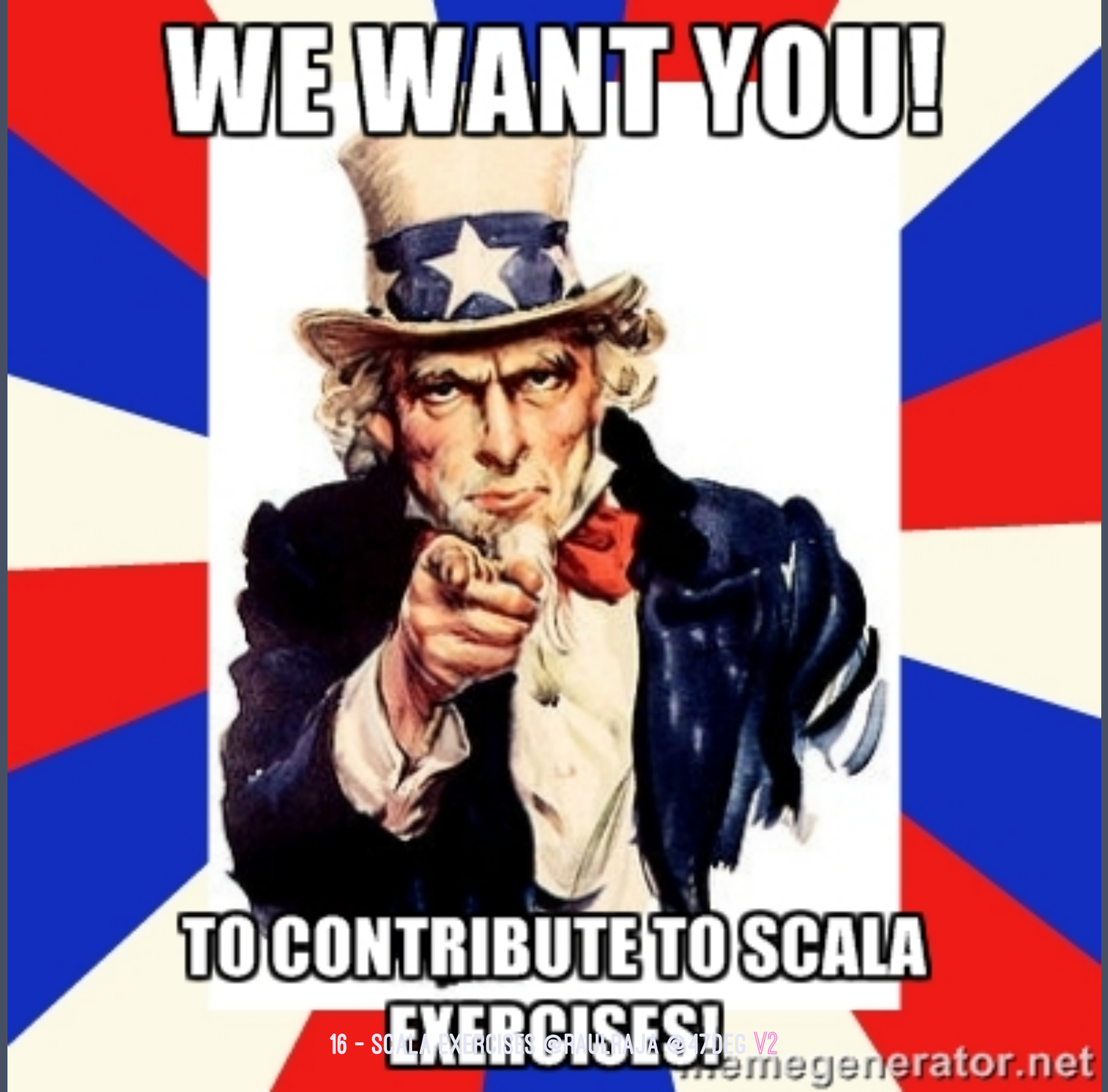
WRITE EXERCISES WITH CODE!

REAL EVALUATION OF EXERCISES

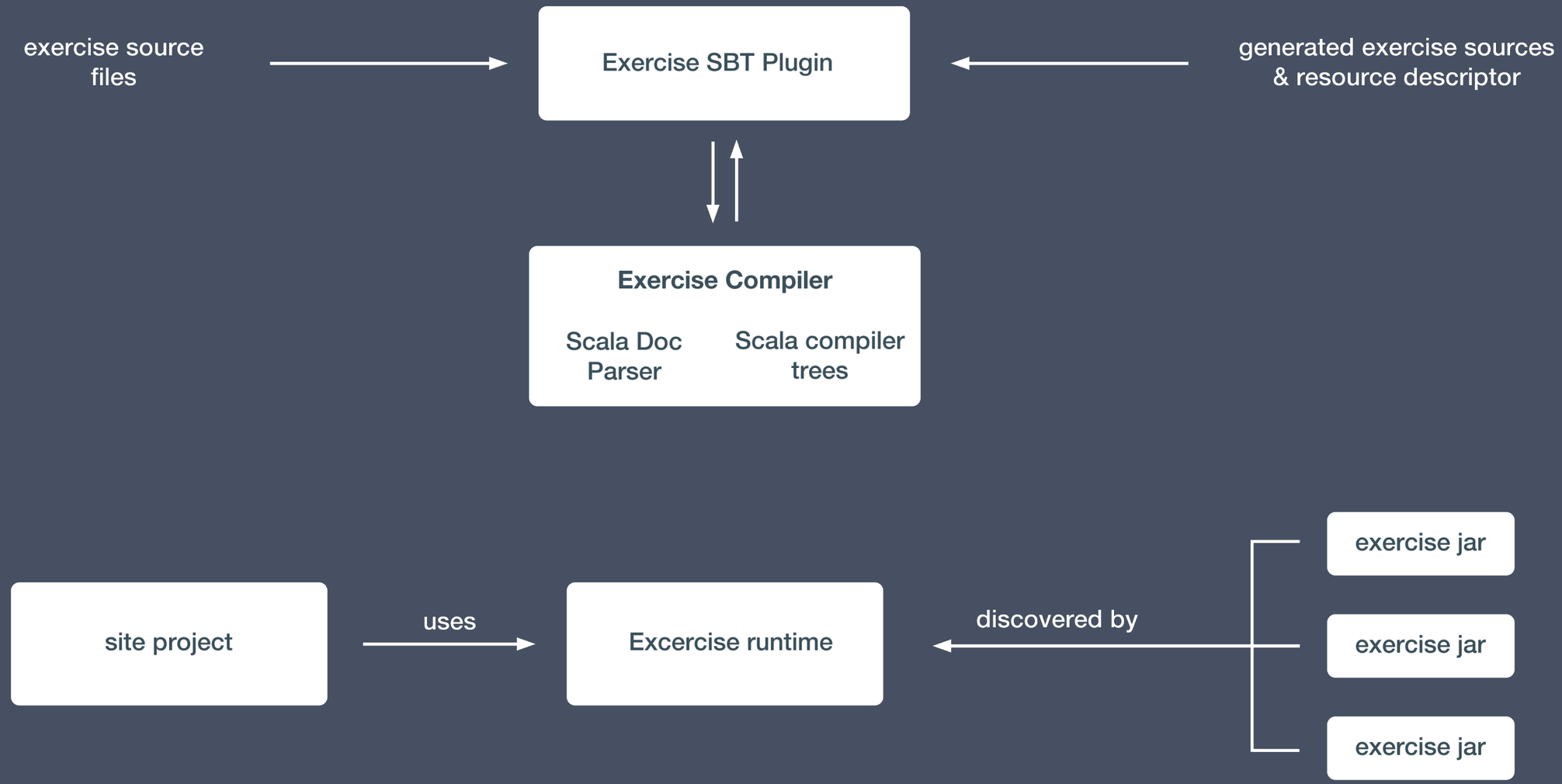


CONTRIBUTORS ARE PROPERLY CREDITED




WE WANT YOU!
TO CONTRIBUTE TO SCALA EXERCISES!
memegenerator.net

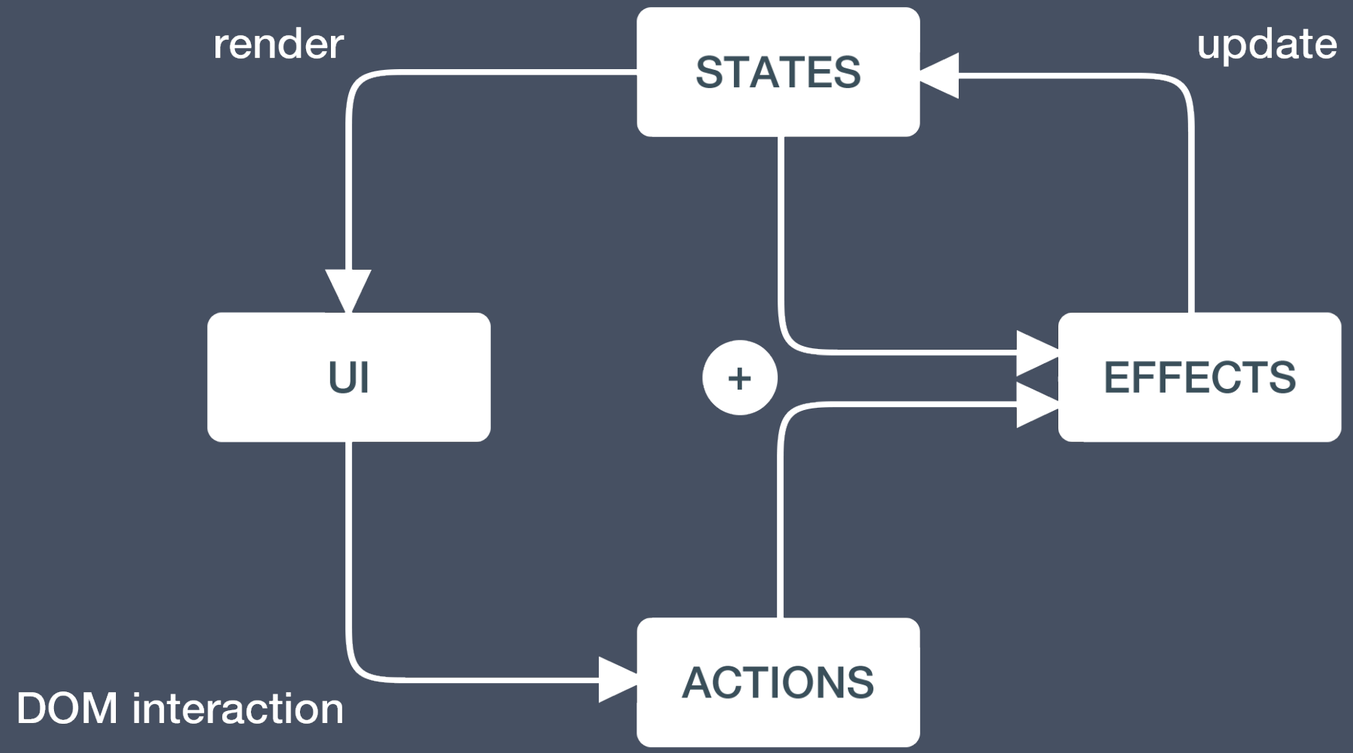
EXERCISES COMPILER ARCHITECTURE



TECHNOLOGIES

- CLIENT [SCALAJS. CATS]
- SERVER [PLAY. CATS. DOOBIE]

CLIENT ARCHITECTURE



BACKEND ARCHITECTURE

```
type App = Coproduct [


]
```

BACKEND ARCHITECTURE

```
type App = Coproduct [
```


```
]
```

BACKEND ARCHITECTURE

```
type App = Coproduct [
```

ExerciseOp
- GetLibraries - GetLibrary - GetSection - Evaluate

,

DBResult
- Success - Failure

```
]
```

BACKEND ARCHITECTURE

```
type App = Coproduct [
```

ExerciseOp
- GetLibraries - GetLibrary - GetSection - Evaluate

,

DBResult
- Success - Failure

,

UserOp
- GetUser - GetUserByLogin - CreateUser - UpdateUser - DeleteUser

```
]
```

BACKEND ARCHITECTURE

```
type App = Coproduct [
```

ExerciseOp
- GetLibraries - GetLibrary - GetSection - Evaluate

,

DBResult
- Success - Failure

,

UserOp
- GetUser - GetUserByLogin - CreateUser - UpdateUser - DeleteUser

,

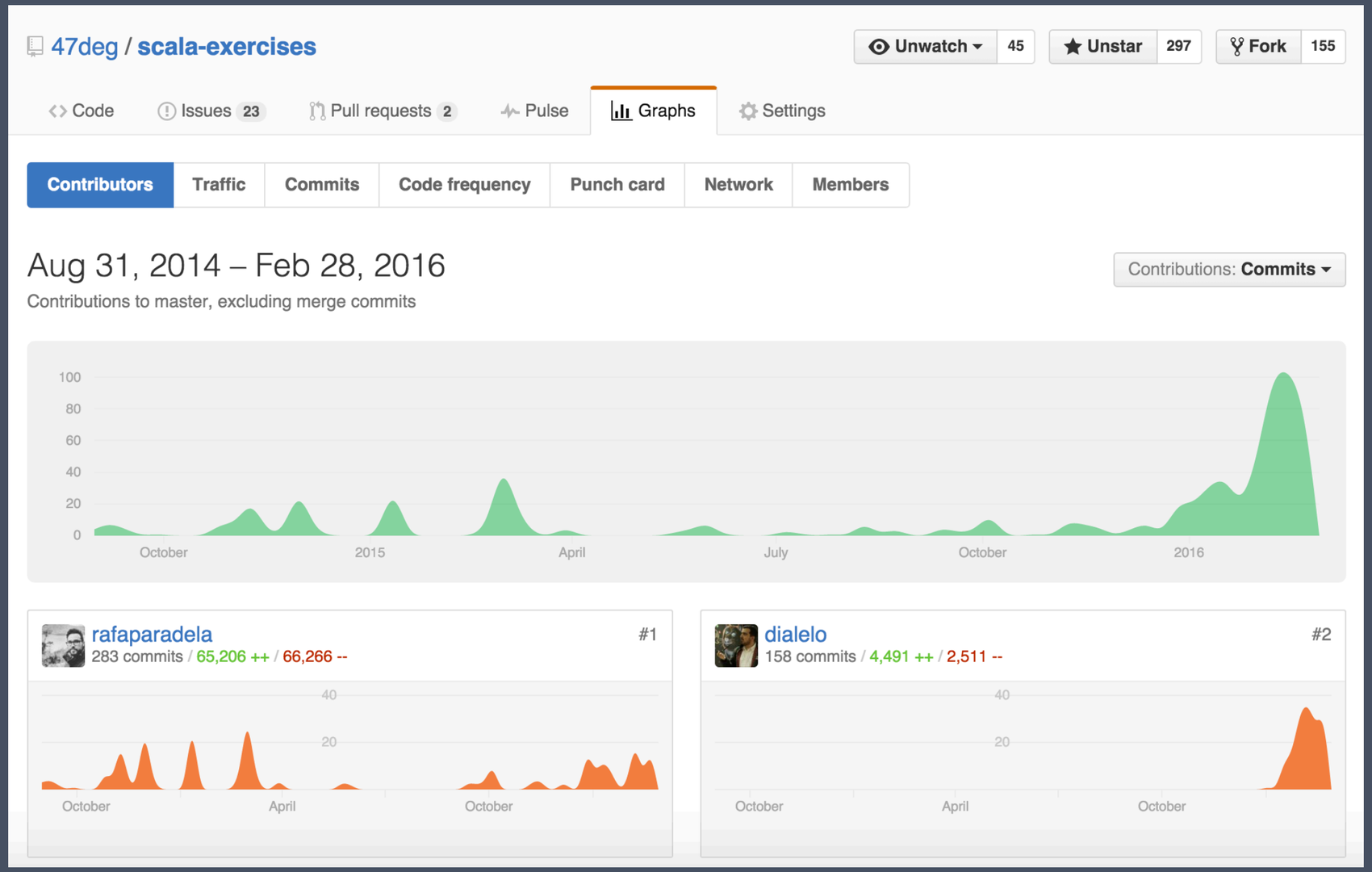
UserProgress
- UpdateProgress

```
]
```

FREE MONADS!

~~FREE~~ MONADAS LIBRES!

WARNING! V2 IS UNDER HEAVY DEVELOPMENT!



ACTIVELY WORKING ON...

- IMPROVED EVALUATION / FORK EVALUATION (FINCH)
- SUPPORT FOR BLOCK STYLE EXERCISES (EX : IMPLEMENT A TYPE CLASS)
- DOCS, CONTRIBUTION GUIDE...
- SPLITTING INTO MULTIPLE REPOSITORIES

WHAT WOULD BE COMING NEXT?

- ALLOW GITHUB ORGANIZATIONS ?
- SUPPORT OTHER EXERCISES FORMAT
- TOP LEVEL DOMAIN / ORG
- IT REALLY IS ENTIRELY UP TO YOU!

THANKS!

@RAULRAJA

@47DEG

- GITHUB : HTTPS://GITHUB.COM/47DEG/SCALA-EXERCISES
- DECK: HTTPS://GITHUB.COM/47DEG/SCALA-EXERCISES-V2-DECK
- V1: HTTP://SCALA-EXERCISES.47DEG.COM