


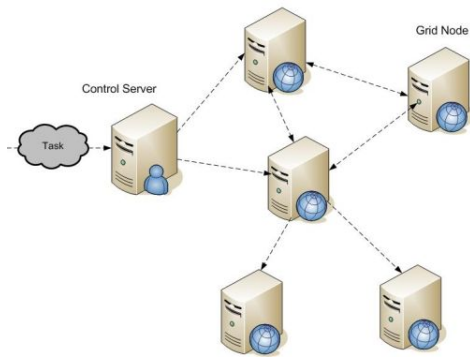


# Spark support

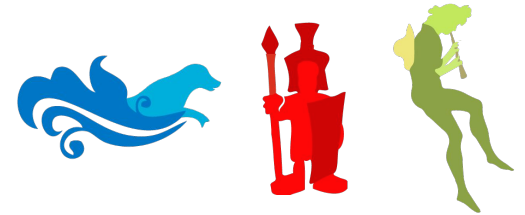
1. Remote execution at Zeppelin
2. Sending jobs to Livy Server
3. User-friendly data representation

*JetBrains spring practice 2018  
Dmitriy Naidanov, Roman Shein, Anton  
Yalyshev*





# Scala



## V. Top Paying Tech

Overview

Developer Profile

### Technology

I. Most Popular Technologies

II. Most Loved, Dreaded, and Wanted

III. Top Tech on Stack Overflow

IV. Trending Tech on Stack Overflow

### V. Top Paying Tech

VI. Correlated Technologies

VII. Development Environments

VIII. Desktop Operating System

Work

Community

Back to top ↕

Top Paying Tech in US

Top Paying Tech Worldwide

Spark \$125,000

Scala \$125,000

Cassandra \$115,000

F# \$115,000

Hadoop \$115,000

Cloud (AWS, GAE, Azure, etc.) \$105,000

Redis \$105,000

Go \$105,000

Clojure \$105,000

# I. Удаленное исполнение кода на Zeppelin


**Zeppelin** Notebook - Interpreter Connected

```
import sys.process._
// sc is an existing SparkContext.
val sqlContext = new org.apache.spark.sql.SQLContext(sc)
val health_dataset = sc.textFile("/Users/nshawa/Downloads/health_data_expenses.csv")
case class Health(year: String, state: String, category: String, funding_src1: String, funding_src2: String, spending: Integer)
val health = health_dataset.map(k=>k.split(",")).map(
  k => Health(k(0),
    k(1),
    k(2),
    k(3),
    k(4),
    k(5).toInt
  )
).toDF()
health.registerTempTable("health_table")

import sys.process._
sqlContext: org.apache.spark.sql.SQLContext = org.apache.spark.sql.SQLContext@7e76cc70
health_dataset: org.apache.spark.rdd.RDD[String] = /Users/nshawa/Downloads/health_data_expenses.csv MapPartitionsRDD[566] at textFile at <console>:182
defined class Health
health: org.apache.spark.sql.DataFrame = [year: string, state: string, category: string, funding_src1: string, funding_src2: string, spending: int]
Took 4 seconds
```

**Xsql** FINISHED

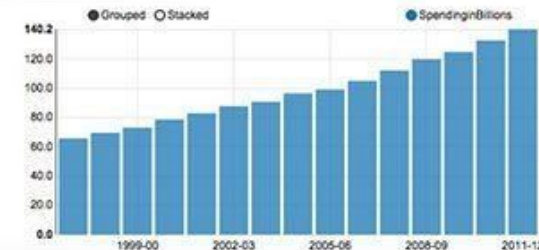
```
select state,sum(spending)/1000 SpendingInBillions from health_table group by state order by SpendingInBillions desc
```



State	Spending (Billions)
NSW	~445.845
VIC	~272.507
QLD	~121.022
WA	~104.221
SA	~90.786
TAS	~75.765
ACT	~72.698
NT	~70.508

**Xsql** FINISHED

```
select year,sum(spending)/1000 SpendingInBillions from health_table group by year order by SpendingInBillions
```



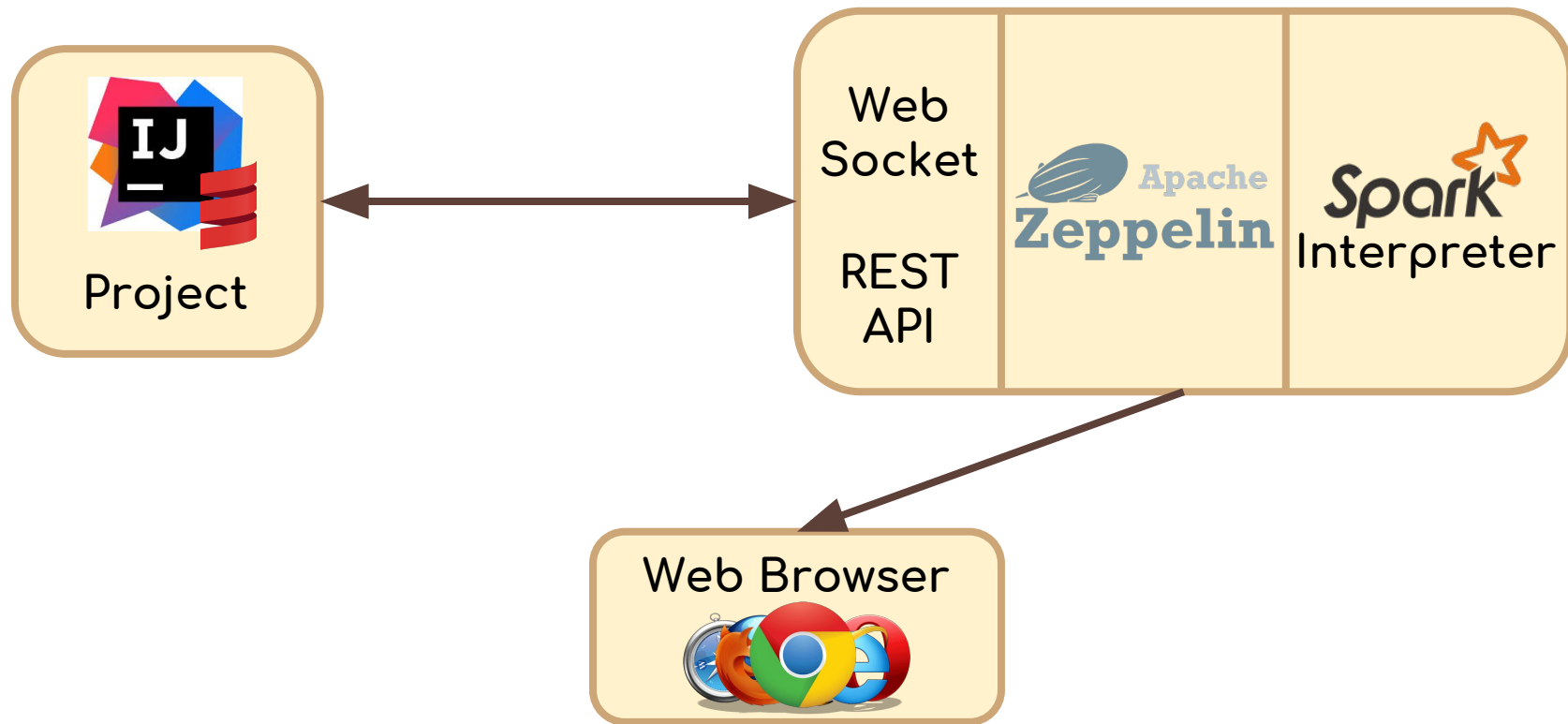
Year	Spending (Billions)
1999-00	~65
2000-01	~70
2001-02	~75
2002-03	~80
2003-04	~85
2004-05	~90
2005-06	~95
2006-07	~100
2007-08	~105
2008-09	~110
2009-10	~115
2010-11	~120
2011-12	~125

**Xsql** FINISHED

```
select category,sum(spending)/1000 SpendingInBillions from health_table group by category order by SpendingInBillions desc
```

category	SpendingInBillions
Public hospitals	445.845
Medical services	272.507
Private hospitals	121.022
Benefit-paid pharmaceuticals	104.221
Dental services	90.786
Community health	75.765
Capital expenditure	72.698
All other medications	70.508
Other health practitioners	51.382

# 1. Удаленное исполнение кода на Zeppelin



# 1. Удаленное исполнение кода на Zeppelin

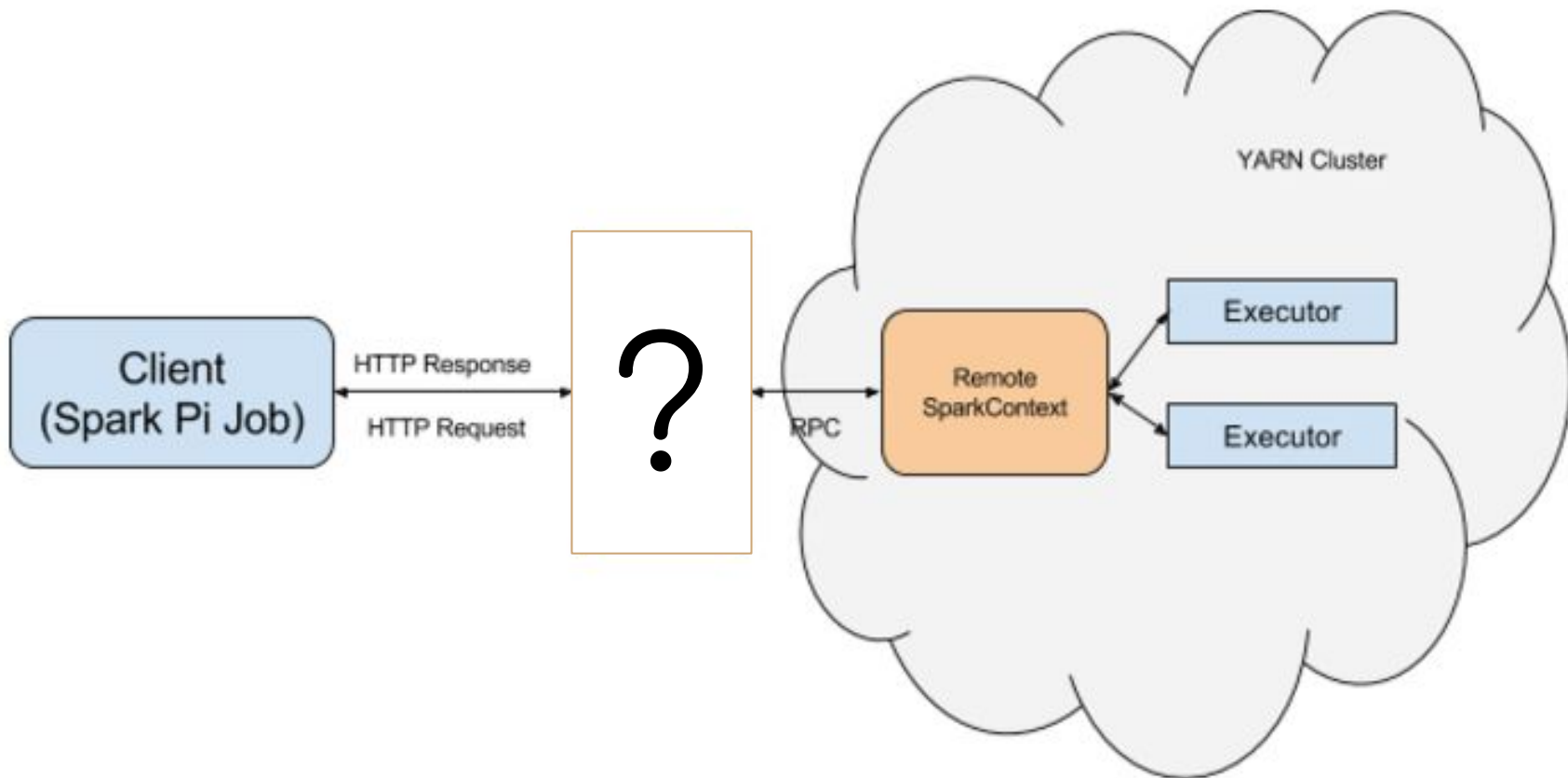
Что делать:

1. Механизм работы ран. конфигураций в IntelliJ IDEA
2. Установка, настройка Apache Zeppelin server
3. Взаимодействие IDEA и Zeppelin через REST API

Что потребуется:

1. Java, Scala
2. Функциональное программирование
3. REST взаимодействие

## 2. Отправка задач на Livy server



## 2. Отправка задач на Livy server

Что делать:

1. Вычислительный кластер
2. Установка, настройка Apache Livy server
3. Механизм работы IDEA с Livy через REST API

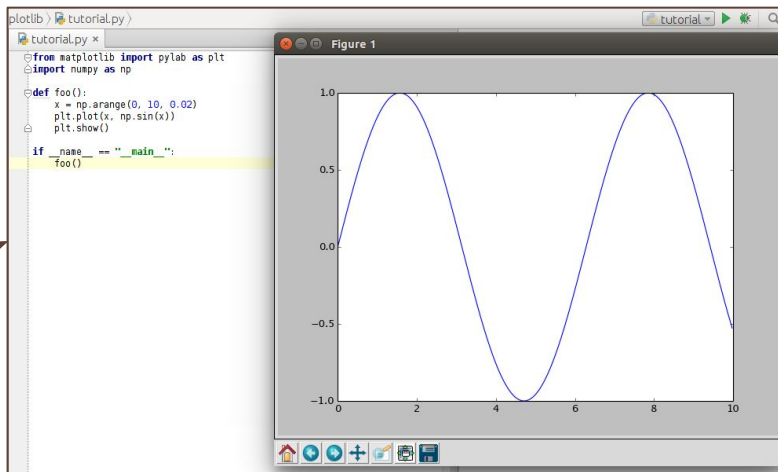
Что потребуется:

1. Java, Scala
2. Функциональное программирование
3. REST взаимодействие



### 3. Визуализация расчётов

- RDD [ T ]
- DataFrame
- Dataset



The screenshot shows a MySQL Database Console window titled 'Database Console MySQL' with a tab for 'test.family'. The table contains the following data:

member_id	name	relation
1	Chloe	mother
2	Harry	father
3	Dylan	brother

# 3. Визуализация расчётов

Что делать:

1. Разобраться с хранением результатов
2. Наиболее удобный вариант визуализации

Что потребуется:

1. Java, Scala
2. Функциональное программирование
3. Опыт с Plotting libs

# References

- Тестовое задание:
  - Удалённое исполнение Zeppelin: <https://stepik.org/lesson/68757>
  - Отправка Job'ов на Livy server: <https://stepik.org/lesson/68758>
  - Графическое отображение: <https://stepik.org/lesson/68759>
- Описание проектов:
  - <https://jetbrains.ru/students/practice/themes/zeppelin-spark-code/>
  - <https://jetbrains.ru/students/practice/themes/spark-jobs-to-apache-server/>
  - <https://jetbrains.ru/students/practice/themes/spark-data-visualization/>
- E-mail: [anton.yalyshev@jetbrains.com](mailto:anton.yalyshev@jetbrains.com)