

Play! Framework ve PostgreSQL Uygulamaları

Şakir Çağlar Toklu

caglartoklu@gmail.com

<http://twitter.com/caglartoklu>

<http://gplus.to/caglartoklu>

Ahmet Yunus Kokulu

yunuskokulu@gmail.com

<http://gplus.to/yunuskokulu>

Biz Kimiz?

- Şakir Çağlar Toklu
 - Proje Yöneticisi, TTG Uluslararası LTD.
- Ahmet Yunus Kokulu
 - Yazılım Geliştirici, TTG Uluslararası LTD.

İçindekiler

- FAMAN ve TTeX Ürünleri
- Prensipler
- Play! Hakkında
- Bir Play! Projesi
- Play! ile veritabanı bağlantısı
- Play! ile veritabanının oluşturulması
- Sequence
- Index
- Miras
- Zamanlı Görevler
- Birim Test Yöntemleri

FAMAN ve TTeX Ürünleri

- FAMAN (Fault Management)
- TTeX (Trouble Ticketing)
- Telekom
- Avrupa Birliği kapsamında, Ericsson ile beraber, KKTC Telefon Dairesi Projesi
- Daha önce Oracle ve ASP.NET

FAMAN ve TteX Verileri

- SNMP Adaptörü
- CORBA Adaptörü
- Web Arayüzü
- Canlı veri

1.1.3.6.1.2.1.1.3.0	43 days, 2:07:52.06
1.1.3.6.1.6.3.1.1.4.1	1.3.6.1.4.1.2352.2.12.0.0.1
1.1.3.6.1.4.1.2352.2.12.1.0.1.1.2	[REDACTED]
1.1.3.6.1.4.1.2352.2.12.1.0.1.1.3	[REDACTED]
1.1.3.6.1.4.1.2352.2.12.1.0.1.1.4	07:dc:04:0c:09:14:24:00
1.1.3.6.1.4.1.2352.2.12.1.0.1.1.5	Proxy to node communication down: [REDACTED]
1.1.3.6.1.4.1.2352.2.12.1.0.1.1.6	1570
1.1.3.6.1.4.1.2352.2.12.1.0.1.1.7	6
1.1.3.6.1.4.1.2352.2.12.1.0.1.1.3	[REDACTED]
1.1.3.6.1.2.1.1.3.0	43 days, 2:07:52.62

Prensipier

- Kendini Tekrar Etme! (DRY)
- En az ayarlama
- Tařınabilirlik
- Uzaktan Müdahele
- Çeviklik

Neden PostgreSQL?

- Proje Gereksinimleri
- Devrim Gündüz
- Geliştirme Ortamı Özgürlüğü
- Platformdan Bağımsızlık
- Kolayca eski sürümlere veya ileri sürümlere geçebilme, uyumluluk testleri
- Veri bütünlüğü
- Lisans: PostgreSQL (MIT+BSD)
- Fiyat

Geliştirme Araçları

- Ubuntu, Linux Mint, CentOS
- BitNami LAPPStack
- BitNami WAPPStack
- Play! Framework
- Eclipse
- Bazaar

Play! Framework Hakkında - 1

- MVC (Model-View-Controller)
- Java EE değil
- Play! 1.2.x, Java, Groovy, Scala
- Play! 2, Scala, Java
- Sıfır paylaşım (Share Nothing)
- Veritabanı bağımsızlığı
- Yeniden derleme ve yükleme
- H2 ve Netty sayesinde hemen çalışmaya başlayabilme



Play! Framework Hakkında - 2

- Web sunucu olarak JBoss Netty
- Hibernate / JPA
- Groovy, şablon dili
- H2 veritabanı

Nesne İlişkilendirici Ne İşe Yarar?

- ORM (Object Relational Mapper)
- Hibernate, NHibernate, LLBLGen Pro, SQLAlchemy
- Kayıtların nesneye yönelik bir dilde ifade edilmesi
 - Kişi Kaydı (SQL) → Kişi Sınıfı (Java)
- Veritabanından model oluşturmak (LLBLGen Pro)
- Modelden veritabanı oluşturmak (Hibernate, Play!)

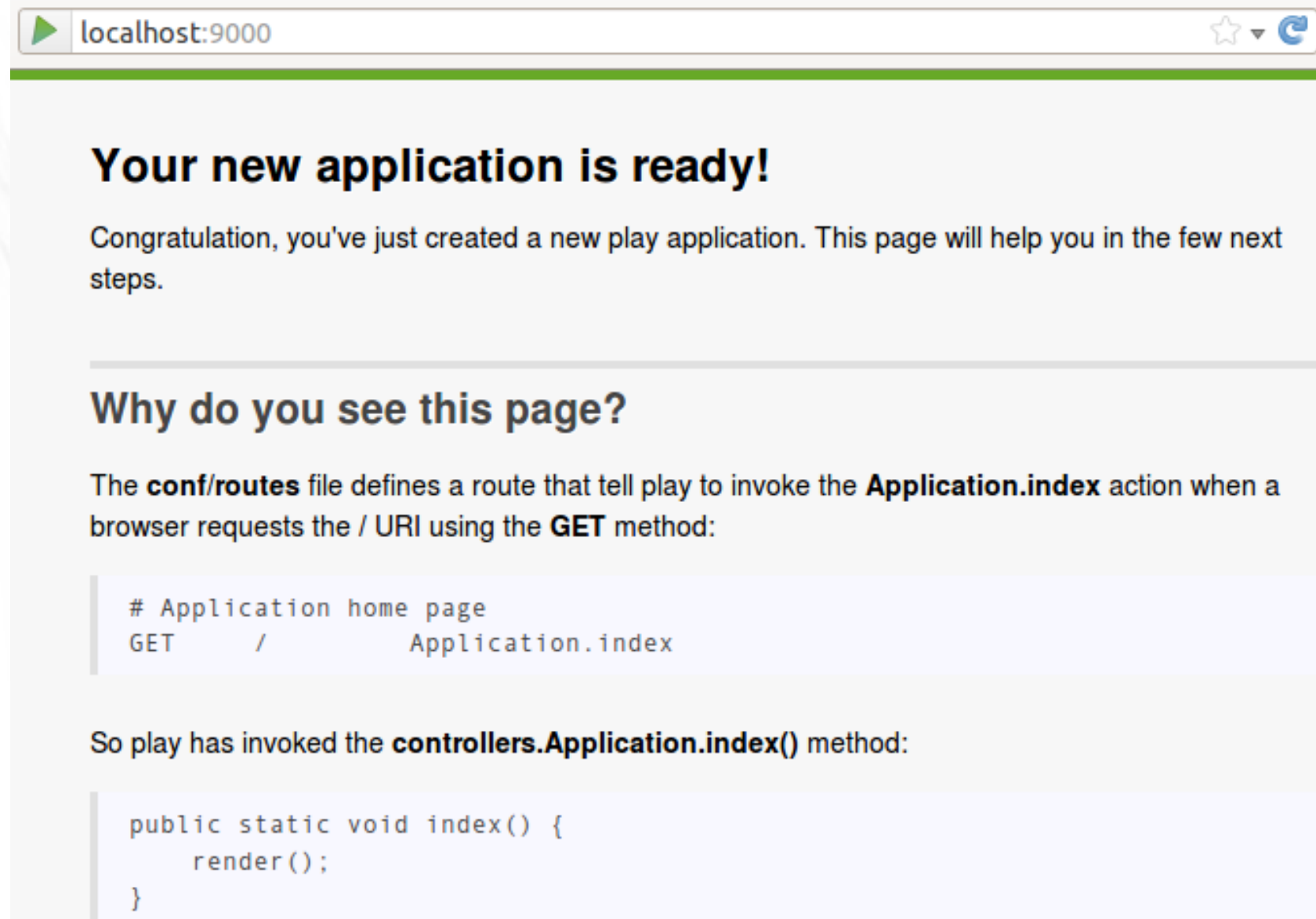
Play! ile Yeni Proje

```
alex@tencere6:~$ play new Proje1  
~  
~  
~    
~  
~  
~ play! 1.2.4, http://www.playframework.org  
~  
~ The new application will be created in /home/alex/Proje1  
~ What is the application name? [Proje1]  
~  
~ OK, the application is created.  
~ Start it with : play run Proje1  
~ Have fun!  
~  
alex@tencere6:~$ cd Proje1  
alex@tencere6:~/Proje1$ ls  
app conf lib public test  
alex@tencere6:~/Proje1$ █
```

Bir Play! Projesinin Anatomisi

```
/home/alex/Proje1/  
/~app/  
| /~controllers/  
| | `Application.java  
| /~models/  
| `~views/  
| | /~Application/  
| | | `index.html  
| | /~errors/  
| | | /-404.html  
| | | `-500.html  
| | `main.html  
/~conf/  
| /-application.conf  
| /-dependencies.yml  
| /-messages  
| `routes  
/~lib/  
/~public/  
| /~images/  
| /~javascripts/  
| | `jquery-1.6.4.min.js  
| `~stylesheets/  
| | `main.css  
/~test/  
| /-Application.test.html  
| /-ApplicationTest.java  
| /-BasicTest.java
```


Play! Projesinin Çalıştırılması - 2



The screenshot shows a web browser window with the address bar displaying 'localhost:9000'. The page content includes a green header bar, a main heading 'Your new application is ready!', a congratulatory message, a section titled 'Why do you see this page?', an explanation of the routing configuration, a code block for the routes file, an explanation of the controller method invocation, and another code block for the controller's index method.

localhost:9000

Your new application is ready!

Congratulation, you've just created a new play application. This page will help you in the few next steps.

Why do you see this page?

The **conf/routes** file defines a route that tell play to invoke the **Application.index** action when a browser requests the / URI using the **GET** method:

```
# Application home page
GET      /           Application.index
```

So play has invoked the **controllers.Application.index()** method:

```
public static void index() {
    render();
}
```

Play! ile PostgreSQL'e Bağlanmak - 1

PostgreSQL 9.1.1 running on localhost:5432 – You are logged in as user "postgres"

 phpPgAdmin:  PostgreSQL?

Create role?

Name	<input type="text" value="user1"/>
Password	<input type="password" value="....."/>
Confirm	<input type="password" value="....."/>
Superuser?	<input checked="" type="checkbox"/>
Create DB?	<input type="checkbox"/>
Can create role?	<input type="checkbox"/>
Inherits privileges?	<input type="checkbox"/>
Can login?	<input checked="" type="checkbox"/>
Connection limit	<input type="text"/>
Expires	<input type="text"/>

Play! ile PostgreSQL'e Bağlanmak - 2

PostgreSQL 9.1.1 running on localhost:5432 – You are logged in as user "user1"

 phpPgAdmin:  PostgreSQL?

Create database?

Name	<input type="text" value="ornek1"/>
Template	<input type="text" value="template1"/>
Encoding	<input type="text" value="UTF8"/>
Collation	<input type="text"/>
Character Type	<input type="text"/>
Comment	<input type="text"/>

Create

Cancel

Play! ile PostgreSQL'e Bağlanmak - 3

```
alex@tencere6:~/Proje1/conf$ ls *.conf
application.conf
alex@tencere6:~/Proje1/conf$
```

```
78 # Database configuration
79 # ~~~~~
80 # Enable a database engine if needed.
81 #
82 # To quickly set up a development database, use either:
83 #   - mem : for a transient in memory database (H2 in memory)
84 #   - fs  : for a simple file written database (H2 file stored)
85 # db=mem
86 #
87 # To connect to a local MySQL5 database, use:
88 # db=mysql://user:pwd@host/database
89 #
90 # To connect to a local PostgreSQL9 database, use:
91 # db=postgres://user:pwd@host/database
92 #
93 db=postgres://user1:password1@127.0.0.1:5432/ornek1
94 #
95 # If you need a full JDBC configuration use the following :
96 # db.url=jdbc:postgresql:database_name
97 # db.driver=org.postgresql.Driver
98 # db.user=root
99 # db.pass=secret
```

Kod Üzerindeki Modellerden Veritabanının Otomatik Oluşturulması - 1

- Kendini Tekrar Etme!
- Temel bir model sınıfı


```
Kisi kisi = new Kisi("Kira", "Yamato");  
kisi.save();
```

```
1 package models;  
2 import play.db.jpa.*;  
3 import javax.persistence.Entity;  
4 import javax.persistence.Column;  
5 import play.data.validation.Required;  
6 @Entity  
7 public class Kisi extends Model {  
8     @Required  
9     @Column(nullable=false, length=100)  
10    public String adi;  
11  
12    @Required  
13    @Column(nullable=false, length=100)  
14    public String soyadi;  
15  
16    @Column(nullable=true)  
17    public Integer yasi;  
18  
19    public Kisi(String adi, String soyadi) {  
20        this.adi = adi;  
21        this.soyadi = soyadi;  
22    }  
23 }
```

Kod Üzerindeki Modellerden Veritabanının Otomatik Oluşturulması - 2

phpPgAdmin: PostgreSQL?: omeek1?: public?: kisi?:

Columns Indexes? Constraints? Triggers?

Column	Type	Not Null	Default	Constraints	
id	bigint	NOT NULL			Browse
adi	character varying(100)	NOT NULL			Browse
soyadi	character varying(100)	NOT NULL			Browse
yasi	integer				Browse

Kod Üzerindeki Modellerden Veritabanınınin Otomatik Oluşturulması - 3

```
1 package models;
2 import play.db.jpa.*;
3 import javax.persistence.Entity;
4 import javax.persistence.Column;
5 import play.data.validation.Required;
6 @Entity
7 public class KullaniciTuru extends Model {
8     @Required
9     @Column(nullable=false, length=20)
10    public String adi ;
11
12    public KullaniciTuru() {
13    }
14 }
```

```
1 package models;
2 import play.db.jpa.*;
3 import javax.persistence.Entity;
4 import javax.persistence.Column;
5 import play.data.validation.Required;
6 import javax.persistence.ManyToOne;
7 import models.KullaniciTuru;
8 @Entity
9 public class Kisi extends Model {
10     @Required
11     @Column(nullable=false, length=100)
12     public String adi;
13
14     @Required
15     @Column(nullable=false, length=100)
16     public String soyadi;
17
18     @Column(nullable=true)
19     public Integer yasi;
20
21     @ManyToOne
22     public KullaniciTuru kullaniciTuru;
23
24     public Kisi(String adi, String soyadi) {
25         this.adi = adi;
26         this.soyadi = soyadi;
27     }
28 }
```

Kod Üzerindeki Modellerden Veritabanının Otomatik Oluşturulması - 4

phpPgAdmin: PostgreSQL?: omeek1?: public?: kisi?:

Columns Indexes? Constraints? Triggers? Rules?



Name	Definition
fk23b054cda03b6	FOREIGN KEY (kullanicituru_id) REFERENCES kullanicituru(id)
kisi_pkey	PRIMARY KEY (id)

Sequence

Name	Start value	Last value	Increment by	Max value	Min value	Cache value
hibernate_sequence	1	5	1	9223372036854775807	1	1

 phpPgAdmin:  PostgreSQL?:  omek1?:  public?:  kullanicituru?:

Insert row?

Column	Type	Format	Null	Value
id	bigint	Expression 		nextval('hibernate_sequence')
adi	character varying(20)	Value 		<input type="text" value="admin"/>

Insert

Insert & Repeat

Cancel

Enable AutoComplete

Index

```
1 package models;
2 import play.db.jpa.*;
3 import javax.persistence.Entity;
4 import javax.persistence.Column;
5 import play.data.validation.Required;
6 import javax.persistence.ManyToOne;
7 import models.KullaniciTuru;
8 import org.hibernate.annotations.Index;
9 @Entity
10 @org.hibernate.annotations.Table(indexes = { @Index(name = "idx_adi_soyadi",
11         columnNames = { "adi", "soyadi" }) }, appliesTo = "Kisi")
12 public class Kisi extends Model {
13     @Required
14     @Column(nullable=false, length=100)
15     public String adi;
16
17     @Required
18     @Column(nullable=false, length=100)
19     public String soyadi;
20
21     @Index(name = "idx_yasi")
22     @Column(nullable=true)
23     public Integer yasi;
24
25     @ManyToOne
26     public KullaniciTuru kullaniciTuru;
27
28     public Kisi(String adi, String soyadi) {
29         this.adi = adi;
30         this.soyadi = soyadi;
31     }
32 }
```




The screenshot shows the phpPgAdmin interface. At the top, there are navigation tabs for 'Columns', 'Indexes?', 'Constraints?', 'Triggers?', and 'Rules?'. The 'Indexes?' tab is selected, displaying a table with the following data:

Name	Definition	Constraints
idx_adi_soyadi	CREATE INDEX idx_adi_soyadi ON kisi USING btree (adi, soyadi)	
idx_yasi	CREATE INDEX idx_yasi ON kisi USING btree (yasi)	
kisi_pkey	CREATE UNIQUE INDEX kisi_pkey ON kisi USING btree (id)	Primary key

Miras

phpPgAdmin: PostgreSQL? omek1? public? kisi?

Columns Indexes? Constraints? Triggers?

Column	Type	Not Null	Default	Constraints	
id	bigint	NOT NULL			Bro
adi	character varying(100)	NOT NULL			Bro
soyadi	character varying(100)	NOT NULL			Bro
yasi	integer				
kullanicituru_id	bigint				
dtype	character varying(31)	NOT NULL			
adresi	character varying(100)				

```
1 package models;
2 import play.db.jpa.*;
3 import javax.persistence.Entity;
4 import javax.persistence.Column;
5 import play.data.validation.Required;
6 @Entity
7 public class Koordinator extends Kisi {
8     @Required
9     @Column(nullable=true, length=100)
10    public String adresi;
11
12    public Koordinator(String adi, String soyadi) {
13        super(adi, soyadi);
14    }
15 }
```

Zamanlı Görevler

- (Scheduled Jobs)
- Zorluklar
- pgAgent
- Play!'in sundukları

```
1 package services;
2
3 import play.jobs.Every;
4 import play.jobs.OnApplicationStart;
5 import play.jobs.Job;
6
7 @Every("10s")
8 // @Every("5min")
9 // @Every("1h")
10 // @OnApplicationStart
11 public class MailAt extends Job {
12     public void doJob() {
13         play.Logger.info("running job!");
14         // do anything here
15     }
16 }
```


Proje Yapısı

```
/~app/  
| |+controllers/  
| |~models/  
| | |/-Kisi.java  
| | |/-Koordinator.java  
| | |`-KullaniciTuru.java  
| |~services/  
| | |`-MailAt.java  
| |`+views/  
|~conf/  
| |/-application.conf  
| |/-dependencies.yml  
| |/-messages  
| |`-routes  
|+lib/  
|+public/  
|+test-result/  
|~test/  
| |/-Application.test.html  
| |/-ApplicationTest.java  
| |/-BasicTest.java  
| |/-data.yml  
| |`-KisiEkleTest.java  
|`+tmp/
```

Son

- Sorular?
- **TENEFFÜS! :)**