



INSTITUTO DE
INFORMÁTICA
UFG



ENGENHARIA
DE SOFTWARE

RELPREV

Procedimento de Implantação - Web server

Versão 1.0

RELPREV	Versão: 1.0
Procedimento de Implantação - Web server	Data: 08/11/2013

Histórico da Revisão

Data	Versão	Descrição	Autor
08/11/2013	1.0	Criação do Documento	Danilo Guimarães

RELPREV	Versão: 1.0
Procedimento de Implantação - Web server	Data: 08/11/2013

Índice

Conteúdo

Objetivos do Documento	4
Procedimento de Implantação	4
Adoção de Maven Profiles	4
Script de criação do banco de dados	5
Usuário e senha do banco de dados	5
Caminho no repositório para geração de builds do produto	6
Produtos gerados pelo build	6
Referências	6

RELPREV	Versão: 1.0
Procedimento de Implantação - Web server	Data: 08/11/2013

Procedimento de Implantação

Objetivos do Documento

Este documento visa explicar paulatinamente o processo de deploy (implantação) dos binários relacionados a parte servidor do projeto RELPREV - Relatório de Prevenção de Acidentes Aeronáuticos da Fábrica de Software do Instituto de Informática da Universidade Federal de Goiás.

Procedimento de Implantação

Segue abaixo os passos do procedimento de implantação do projeto RELPREV:

Adoção de Maven Profiles

Por se tratar de projetos Maven, todo o "ciclo de vida" do build é inteiramente gerenciado pelo framework citado. Bastando apenas definir/escolher os *profiles* necessários para o tipo de build pretendido.

De início possuímos quatro profiles distintos, para dois propósitos:

- **Ambiente**
 - *dev*
 - *prod*
- **Banco de dados**
 - *postgres*
 - *mariadb*

O propósito dos dois primeiros profiles (*dev* e *prod*) é o de definir configurações sobre em qual ambiente o build se destinará. Por padrão, o *prod* é sempre ativado.

Já os dois últimos profiles (*postgresql* e *mariadb*) tem como propósito definir para qual solução de banco de dados o build será gerado. Por padrão, o *mariadb* é sempre ativado.

O ambiente de produção da Fábrica de Software do INF-UFG utilizará a solução MariaDB, portanto o profile homônimo deverá ser ativado.

Segue abaixo um exemplo de declaração dos quatros profiles*:

```
<profile>
  <id>dev</id>
  <properties>
    <log4j.level>debug</log4j.level>

    <!-- hibernate properties -->
    <hibernate.show.sql>true</hibernate.show.sql>
    <hibernate.format.sql>true</hibernate.format.sql>
    <hibernate.hbm2ddl.auto>update</hibernate.hbm2ddl.auto>
    <hibernate.use_sql_comments>true</hibernate.use_sql_comments>
    <hibernate.generate_statistics>true</hibernate.generate_statistics>
  </properties>
</profile>
<profile>
  <id>prod</id>
  <activation>
    <activeByDefault>true</activeByDefault>
  </activation>
</profile>
```

RELPREV	Versão: 1.0
Procedimento de Implantação - Web server	Data: 08/11/2013

```

    </activation>
    <properties>
        <log4j.level>warn</log4j.level>

        <!-- hibernate properties -->
        <hibernate.show.sql>>false</hibernate.show.sql>
        <hibernate.format.sql>>false</hibernate.format.sql>
        <hibernate.hbm2ddl.auto>update</hibernate.hbm2ddl.auto>
        <hibernate.use_sql_comments>>false</hibernate.use_sql_comments>
        <hibernate.generate_statistics>>false</hibernate.generate_statistics>
    </properties>
</profile>
<profile>
    <id>postgresql</id>
    <properties>
        <log4j.level>DEBUG</log4j.level>

        <jdbc.groupId>org.postgresql</jdbc.groupId>
        <jdbc.artifactId>postgresql</jdbc.artifactId>
        <jdbc.version>9.2-1003-jdbc4</jdbc.version>
        <jdbc.user>postgres</jdbc.user>
        <jdbc.password>postgres</jdbc.password>
        <jdbc.driver>org.postgresql.Driver</jdbc.driver>
        <jdbc.url>jdbc:postgresql://localhost:5432/relprev</jdbc.url>

        <hibernate.dialect>org.hibernate.dialect.PostgreSQL82Dialect</hibernate.dialect>
    </properties>
</profile>
<profile>
    <id>mariadb</id>
    <activation>
        <activeByDefault>true</activeByDefault>
    </activation>
    <properties>
        <jdbc.groupId>org.jumpmind.symmetric.jdbc</jdbc.groupId>
        <jdbc.artifactId>mariadb-java-client</jdbc.artifactId>
        <jdbc.version>1.1.1</jdbc.version>
        <jdbc.user>relprev</jdbc.user>
        <jdbc.password>relprev</jdbc.password>
        <jdbc.driver>org.mariadb.jdbc.Driver</jdbc.driver>
        <jdbc.url>jdbc:mariadb://200.137.197.238/relprev</jdbc.url>

        <hibernate.dialect>org.hibernate.dialect.MySQL5Dialect</hibernate.dialect>
    </properties>
</profile>

```

** parâmetros, valores e estrutura sujeitos a alterações.*

Script de criação do banco de dados

Por padrão, a aplicação já irá sincronizar o schema do banco de dados conforme descrito nas entidades de domínio.

Ainda assim, para que o schema seja criado basta definir o parâmetro `<hibernate.hbm2ddl.auto>` com valor "update" (sem aspas) nos profiles `dev` ou `prod`. Lembrando mais uma vez que já é o padrão.

Usuário e senha do banco de dados

Os valores para usuário e senha do banco são definidos, respectivamente, nos parâmetros `<jdbc.user>`

RELPREV	Versão: 1.0
Procedimento de Implantação - Web server	Data: 08/11/2013

e `<jdbc.password>` dos perfis `postgresql` OU `mariadb`.

Caminho no repositório para geração de builds do produto

O caminho dentro do repositório é onde se encontra o `pom.xml` do projeto principal, que no caso está situado sob a URL <https://relprev-022013.googlecode.com/svn/trunk/src/relprev>.

Produtos gerados pelo build

Como resultado do build, serão produzidos dois artefatos: `services.war` e `relprev.war`.

Tais artefatos devem ser implantados no servidor de aplicações do ambiente da Fábrica (Glassfish).

Referências

Maven: Introduction to Profiles, disponível em <http://maven.apache.org/guides/introduction/introduction-to-profiles.html>, acessado em 08/11/2013.