

JpaResourceLoader

The following is an example of a ResourceLoader that uses an EntityManager and NamedQuery to retrieve templates:

```
package mysite.velocity;

import java.io.BufferedInputStream;
import java.io.ByteArrayInputStream;
import java.io.InputStream;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.NoResultException;
import javax.persistence.Query;

import org.apache.commons.collections.ExtendedProperties;
import org.apache.commons.lang.StringUtils;
import org.apache.velocity.exception.ResourceNotFoundException;
import org.apache.velocity.runtime.resource.Resource;
import org.apache.velocity.runtime.resource.loader.ResourceLoader;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

/**
 * JPA EntityManager ResourceLoader implementation.
 *
 * @author David Pedowitz
 */
public class EntityManagerResourceLoader extends ResourceLoader {
    private static final Logger LOGGER = LoggerFactory
        .getLogger(EntityManagerResourceLoader.class);
    private EntityManagerFactory entityManagerFactory;
    private String templateNamedQuery;

    public void init(ExtendedProperties configuration) {
        templateNamedQuery = StringUtils.trimToNull(configuration
            .getString("resource.templateNamedQuery"));

        if (templateNamedQuery == null) {
            throw new RuntimeException("No templateNamedQuery defined.");
        } else if (entityManagerFactory == null) {
            throw new RuntimeException("No entityManagerFactory defined.");
        }

        LOGGER.debug("Configured entity manager resource loader with "
            + "templateNamedQuery:{} and entityManagerFactory:{}",
            templateNamedQuery, entityManagerFactory);
    }

    public long getLastModified(Resource resource) {
        return 0;
    }

    public InputStream getResourceStream(String name)
        throws ResourceNotFoundException {
        LOGGER.debug("Retrieve resource for name:{}", name);
        EntityManager entityManager = entityManagerFactory
            .createEntityManager();

        try {
            Query query = entityManager.createNamedQuery(templateNamedQuery);
            query.setParameter(1, name);
            String template = (String) query.getSingleResult();
            return new BufferedInputStream(new ByteArrayInputStream(template
                .getBytes()));
        } catch (NoResultException ex) {
            throw new ResourceNotFoundException("No resource found for name:"
                + name);
        } finally {
            entityManager.close();
        }
    }
}
```

```

    }
}

public boolean isSourceModified(Resource resource) {
    return false;
}

public void setEntityManagerFactory(
    EntityManagerFactory entityManagerFactory) {
    this.entityManagerFactory = entityManagerFactory;
}
}

```

I used Spring to configure the VelocityEngine and ResourceLoader like so:

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="http://www.springframework.org/schema
/context"
    xmlns:p="http://www.springframework.org/schema/p" xmlns:security="http://www.springframework.org/schema
/security"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
        http://www.springframework.org/schema/tx
        http://www.springframework.org/schema/tx/spring-tx-2.5.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-2.5.xsd
        http://www.springframework.org/schema/security
        http://www.springframework.org/schema/security/spring-security-2.0.xsd">

... contents omitted ...

<bean id="velocityEngine"
    class="org.springframework.ui.velocity.VelocityEngineFactoryBean">
    <property name="velocityPropertiesMap">
        <map>
            <entry key="resource.loader" value="em" />
            <entry key="em.resource.loader.instance" value-ref="entityManagerResourceLoader" />
            <entry key="em.resource.loader.resource.templateNamedQuery"
                value="findEmailTemplateBodyByName" />
        </map>
    </property>
</bean>

<bean id="entityManagerResourceLoader" class="mysite.velocity.EntityManagerResourceLoader"
    p:entityManagerFactory-ref="entityManagerFactory" />

</beans>

```

Enjoy!