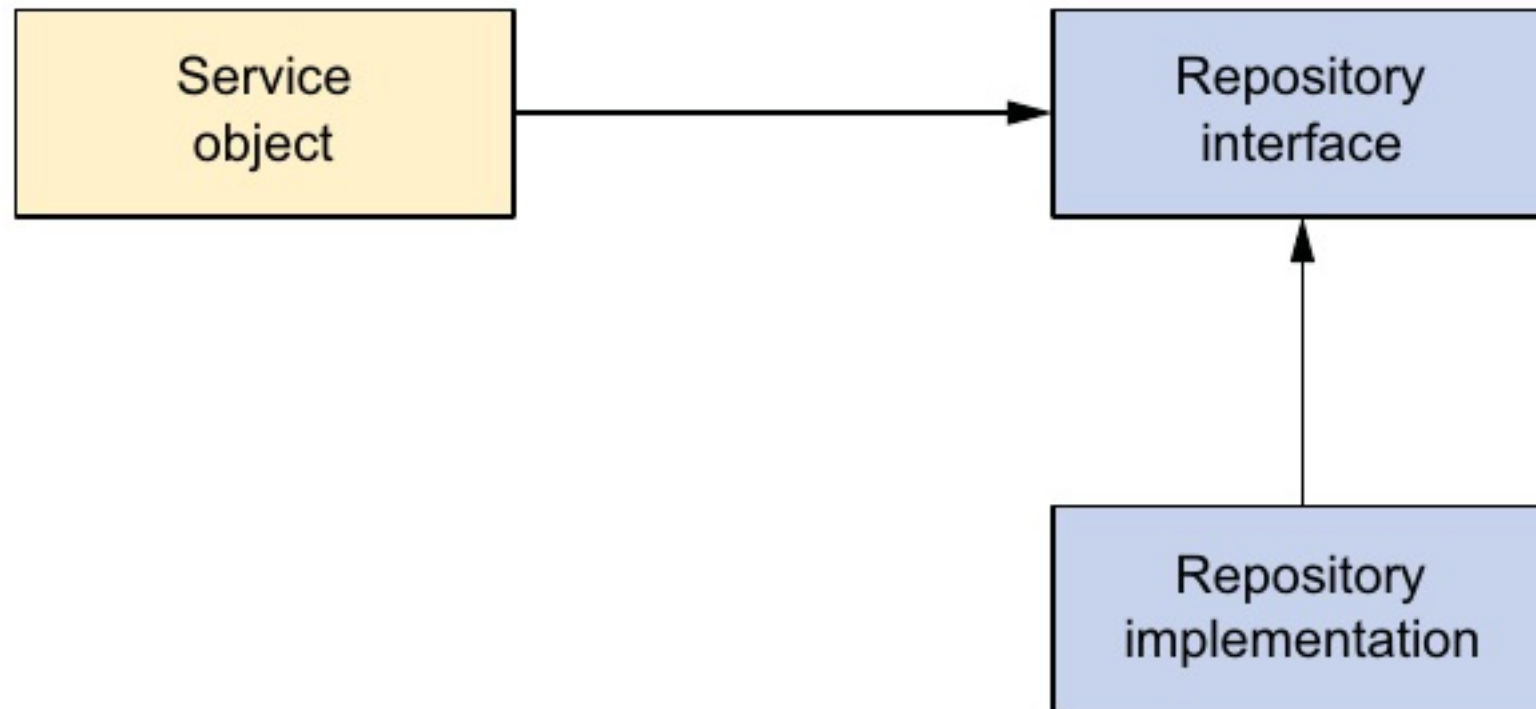
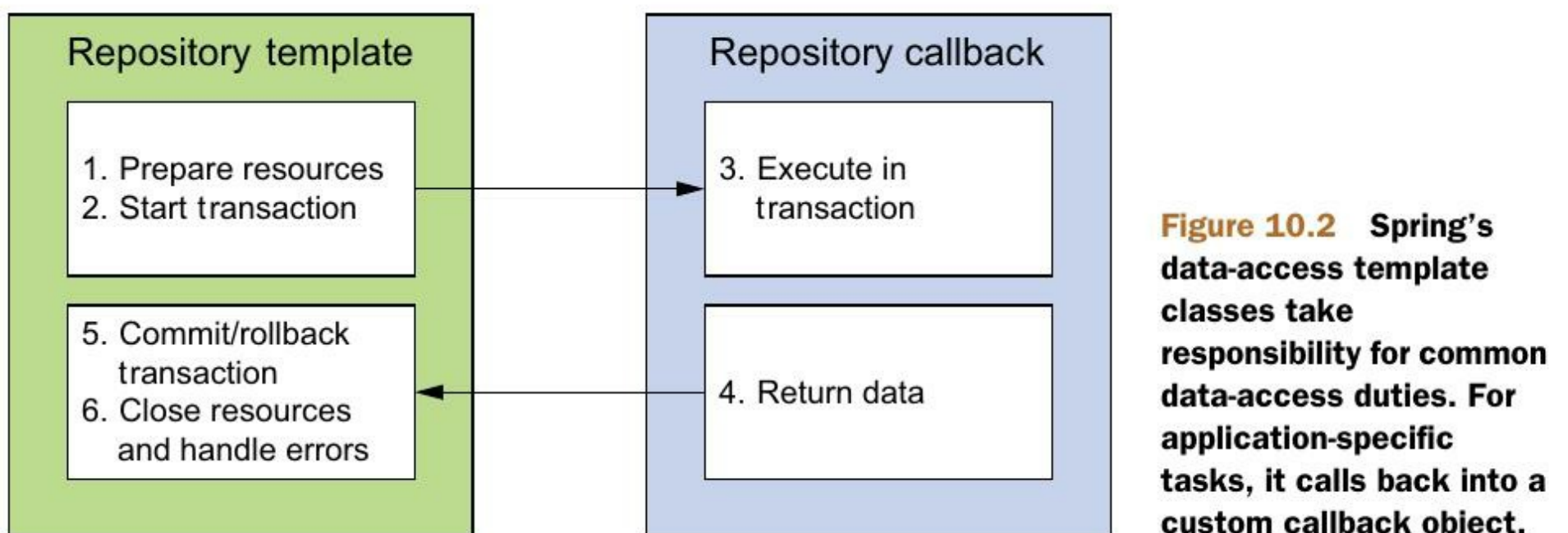


# JDBC in spring

- Слой доступа к данным отделен от других частей через использование интерфейсов:



- Шаблоны содержат стандартный повторяющийся код, каллбеки содержат переменные части:



**Figure 10.2** Spring's data-access template classes take responsibility for common data-access duties. For application-specific tasks, it calls back into a custom callback object.

**Table 10.2 Spring comes with several data-access templates, each suitable for a different persistence mechanism.**

Template class ( <code>org.springframework.*</code> )	Used to template . . .
<code>jca.cci.core.CciTemplate</code>	JCA CCI connections
<code>jdbc.core.JdbcTemplate</code>	JDBC connections
<code>jdbc.core.namedparam.NamedParameterJdbcTemplate</code>	JDBC connections with support for named parameters
<code>jdbc.core.simple.SimpleJdbcTemplate</code>	JDBC connections, simplified with Java 5 constructs (deprecated in Spring 3.1)
<code>orm.hibernate3.HibernateTemplate</code>	Hibernate 3.x+ sessions
<code>orm.ibatis.SqlMapClientTemplate</code>	iBATIS SqlMap clients
<code>orm.jdo.JdoTemplate</code>	Java Data Object implementations
<code>orm.jpa.JpaTemplate</code>	Java Persistence API entity managers

- Spring позволяет использовать различные пулы соединений а так же искать их в JNDI.
- Запустим встроенную БД H2, выполним указанные скрипты и вернем DataSource к этой бд:

```
@Bean
public DataSource dataSource() {
    return new EmbeddedDatabaseBuilder()
        .setType(EmbeddedDatabaseType.H2)
        .addScript("classpath:schema.sql")
        .addScript("classpath:test-data.sql")
        .build();
}
```

