which design pattern is used in spring

* Proxy - used heavily in [AOP](http://static.springframework.org/spring/docs/2.5.x/reference/aop.html#aop-understanding-aop-proxies), and [remoting](http://static.springframework.org/spring/docs/2.5.x/reference/remoting.html).
* Singleton - beans defined in spring config files are singletons by default.
* Template method - used extensively to deal with boilerplate repeated code (such as closing connections cleanly, etc..). For example [JdbcTemplate](http://static.springframework.org/spring/docs/2.5.x/reference/jdbc.html#jdbc-JdbcTemplate), [JmsTemplate](http://static.springframework.org/spring/docs/2.5.x/reference/jms.html#jms-jmstemplate), [JpaTemplate](http://static.springframework.org/spring/docs/2.5.x/reference/orm.html#orm-jpa-template).

**Q17.** **Name some of the Design Patterns used in the Spring Framework?**

* Singleton **Pattern**: Singleton-scoped beans.
* Factory **Pattern**: Bean Factory classes.
* Prototype **Pattern**: Prototype-scoped beans.
* Adapter **Pattern**: **Spring** Web and **Spring** MVC.
* Proxy **Pattern**: **Spring** Aspect Oriented Programming support.

 @Service is a specialisation of @Component to indicate a component in your service layer.   
  
Other specialisations of @Component are   
  
- @Repository to indicate a DAO   
- @Controller in the presentation layer

| @Component | generic stereotype for any Spring-managed component |   
| @Repository| stereotype for persistence layer |   
| @Service | stereotype for service layer |   
| @Controller| stereotype for presentation layer (spring-mvc) |

***@Component***: It is a basic auto component scan annotation, it indicates annotated class is a auto scan component.  
  
***@Repository***: You need to use this annotation with in the persistance layer, which acts like database repository.  
  
***@Service***: It indicates annotated class is a Service component in the business layer.  
  
***@Controller***: Annotated class indicates that it is a controller components, and mainly used at presentation layer.

 @Component is a generic stereotype for any Spring-managed component or bean.

 @Repository is a stereotype for persistence layer.

 @Service is a stereotype for the service layer.

 @Controller is a stereotype for presentation layer (spring-MVC).  
  
Read more: <https://javarevisited.blogspot.com/2017/11/difference-between-component-service.html#ixzz5MLQPlMOC>

@PathVariable can be used in any RequestMethod

@RequestParam annotation:

* **defaultValue** – This is the default value as a fallback mechanism if request is not having the value or it is empty.
* **name** – Name of the parameter to bind
* **required** – Whether the parameter is mandatory or not. If it is true, failing to send that parameter will fail.
* **value** – This is an alias for the name attribute

[**@RequestParam**](https://www.dineshonjava.com/requestparam-annotation-in-spring-mvc-with-example/)

It is used to get the request parameters. @RequestParam automatically binds the request parameters to the arguments of your handler method. It also provides auto type conversion for some standard type like int, long, float, string, date etc.

**Look at the following request URL:**

***http://localhost:8080/tutorials/bookmark/?site=dineshonjava&id=200***

## @PathVariable

It is used to pass parameter along with the url, sometimes we need to pass parameters along with the url to get the data. Spring MVC provides support for customizing the URL in order to get data. To achieving this purpose @PathVariable annotation is used in Spring mvc framework.

**Look at the following request URL:**  
 ***http://localhost:8080/tutorials/bookmark/100?site=dineshonjava&id=200***

Consutroctor

1. <bean id="e" **class**="com.javatpoint.Employee">
2. <constructor-arg value="10" type="int"></constructor-arg>
3. </bean>
5. </beans>

Setter

1. <bean id="obj" **class**="com.javatpoint.Employee">
2. <property name="id">
3. <value>20</value>
4. </property>
5. <property name="name">
6. <value>Arun</value>
7. </property>
8. <property name="city">
9. <value>ghaziabad</value>
10. </property>
12. </bean>