Register permanent view

```
package integration.test.mvc;

import org.springframework.context.annotation.Configuration;
import
org.springframework.web.servlet.config.annotation.ViewControllerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration
public class MvcConfig implements WebMvcConfigurer {

	@Override
	public void addViewControllers(ViewControllerRegistry registry) {
		registry.addViewController("/").setViewName("index");
	}

}
```

In the provided code snippet, the `MvcConfig` class is a configuration class that implements the `WebMvcConfigurer` interface. This configuration class is used to customize the behavior of Spring MVC, specifically for handling view controllers.

Let's break down what this code does:

1. **`@Configuration` Annotation**: Indicates that the `MvcConfig` class is a configuration class and should be processed by the Spring container during application startup.

2. **`implements WebMvcConfigurer`**: By implementing the `WebMvcConfigurer` interface, the `MvcConfig` class can override methods provided by the interface to customize the Spring MVC configuration.

3. **`addViewControllers` Method Override**: Overrides the `addViewControllers` method from the `WebMvcConfigurer` interface. This method is used to register simple mappings between URLs and view names without the need for a controller.

4. **`registry.addViewController("/").setViewName("index")`**: Configures a mapping for the root URL ("/") to a specific view name ("index"). When a request is made to the root URL, Spring MVC will render the view associated with the "index" view name.

   - `registry.addViewController("/").setViewName("index")` creates a mapping

for the root URL ("/") to a view controller.
   - `.setViewName("index")` sets the view name to "index". This means that when a request is made to "/", Spring MVC will render the view with the logical name "index". The actual view resolved by the view resolver configured in the application context will be rendered.

In summary, the `MvcConfig` class with the `addViewControllers` method is used to configure a simple view controller that maps the root URL ("/") to a specific view name ("index"). This approach is often used for rendering static or simple views without the need for a dedicated controller.