

Environment abstraction in Spring

Environment Abstraction is part of Spring Container that models two key aspect of application environment:

- ▶ Profiles
- ▶ Properties

Environment Abstraction is represent on code level by classes that implements `Environment` interface. This interface allows you to resolve properties and also to list profiles. You can receive reference to class that implements `Environment` by calling `EnvironmentCapable` class, implemented by `ApplicationContext`. Properties can also be retrieved by using `@Value("${...}")` annotation.

Environment Abstraction role in context of profiles is to determine which profiles are currently active, and which are activated by default.

Environment Abstraction role in context of properties is to provide convenient, standardized and generic service that allows to resolve properties and also to configure property sources. Properties may come from following sources:

- ▶ Properties Files
- ▶ JVM system properties
- ▶ System Environment Variables
- ▶ JNDI
- ▶ Servlet Config
- ▶ Servlet Context Parameters

Default property sources for standalone applications are configured in `StandardEnvironment`, which includes JVM system properties and System Environment Variables. When running Spring Application in Servlet Environment, property sources will be configured based on `StandardServletEnvironment`, which additionally includes Servlet Config and Servlet Context Parameters, optionally it might include `JndiPropertySource`.

To add additional properties files as property sources you can use `@PropertySource` annotation.

