Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

SPI
Service Provider Interface

Java SPIs
(Service Provider Interfaces)

Prof. Dr. : **Patrick Fuhrmann**

Name: **Mohamed Ali Mohamed Khalil Ali**

Matrikel-Nr.: **575452**

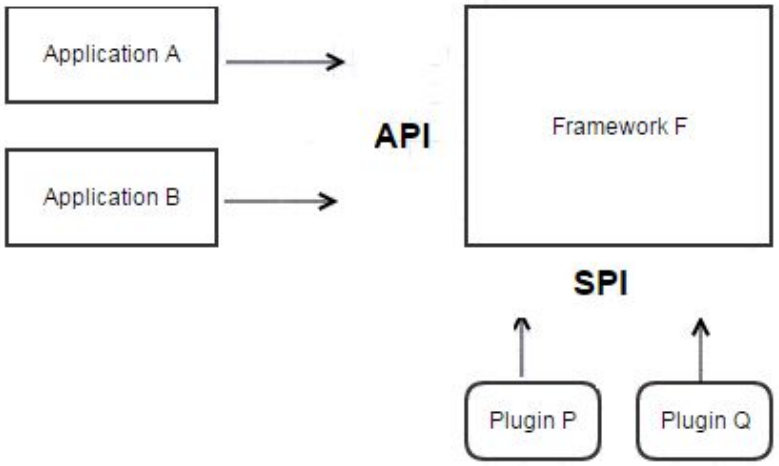
Inhaltsverzeichnis

Übersicht

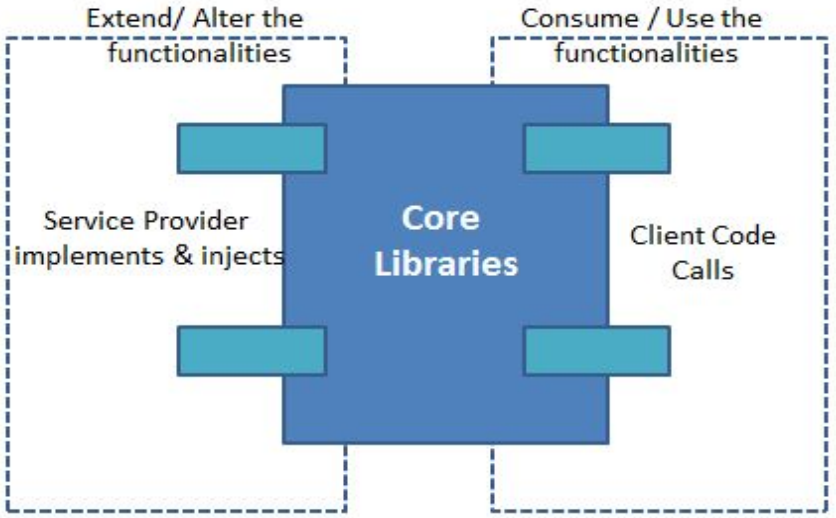
- Java 6 hat eine Funktion zum Erkennen und Laden von Implementierungen eingeführt, die einer bestimmten Schnittstelle entsprechen: Service Provider Interface (**SPI**).

SPI vs API

- **SPI** ist die Beschreibung von Klassen, Schnittstellen oder Methoden, die Sie erweitern und implementieren, um ein Ziel zu erreichen.



- **API** ist die Beschreibung von Klassen, Schnittstellen oder Methoden, die Sie aufrufen und verwenden, um ein Ziel zu erreichen.

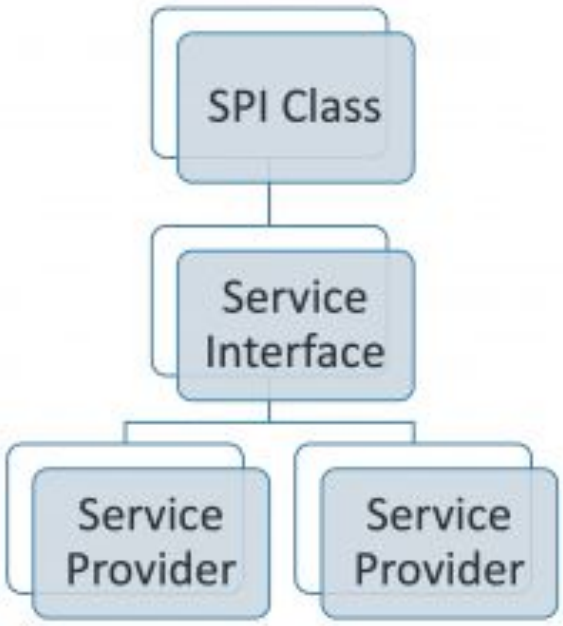


Java-SPI-Hauptkomponenten

- Service Provider Interface.
- Service Providers.
- SPI Configuration File.
- Service Loader.

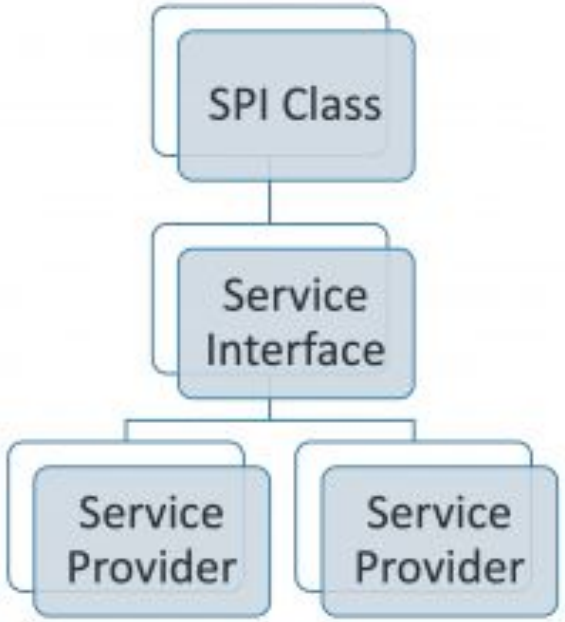
Service Provider Interface

- Eine Schnittstelle oder abstrakte Klasse, die den Vertrag für die Implementierungsklassen der "Service Provider" definiert.



Service Providers

- Die Implementierungsklassen, die die Dienste tatsächlich bereitstellen.



SPI Configuration File

- Eine spezielle Datei, die die Logik für die Suche nach den Implementierungen der "**Service**" bereitstellt.
- Der Dateiname muss im Verzeichnis **META-INF/services** vorhanden sein.
- Der Dateiname sollte genau mit dem voll-qualifizierten Namen der Schnittstelle des Dienstanbieters übereinstimmen.
- Jede Zeile in der Datei enthält die Details einer "**Implementation Service Class**", wiederum den voll-qualifizierten Namen der "Service Provider" klasse.

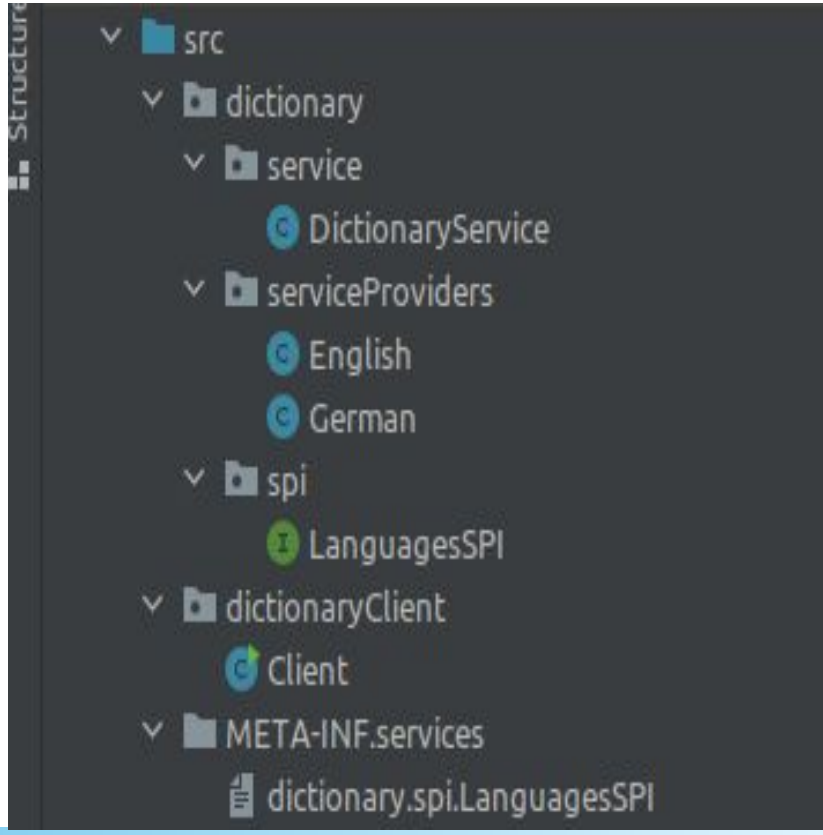
Service Loader

- Die Java SPI-Hauptklasse, die zum Laden der Dienste für eine Dienstanbieterschnittstelle (SPI) verwendet wird.
- Es gibt verschiedene Utility-Methoden im ServiceLoader, um bestimmte Implementierungen zu erhalten, sie zu durchlaufen oder die Dienste erneut zu laden.

Wörterbuch Beispiel

Dies ist ein schönes Beispiel für ein Wörterbuch, das jede Sprache unterstützt, die der Kunde über SPI hinzufügen möchte.

- **Projektstruktur**:
  - DictionaryService (Singelton).
  - LanguagesSPI (SPI).
  - English/German (SPs).
  - META-INF.services (cfg).
  - Client (main).



Service Provider Interface (SPI)

- LanguagesSPI ist unsere “Service Provider Interface”.
- SPI muss bei allen “Service Providers” implementiert werden.
- Es enthält eine Methode “getDifinition”.

```
public interface LanguagesSPI {
    String getDefinition(String word);
}
```

Service

- Methode “getInstance” (Singleton).
- Laden von Implementierungen der “LanguageSPI” mittels ServiceLoader.

```
public class DictionaryService {
    private static DictionaryService dictionaryService;
    private ServiceLoader loader;

    private DictionaryService() { this.loader = ServiceLoader.load(LanguagesSPI.class); }

    public static DictionaryService getInstance() {
        if(DictionaryService.dictionaryService == null)
            return new DictionaryService();
        else
            return DictionaryService.dictionaryService;
    }
```

Service

- Iterates auf alle von ServiceLoader geladene SPI-Implementierungen.
- Es versucht die Difinition des übergegebenen Wort in allen Implementierungen zu finden und gibt das zurück, ansonst gibt "null" zurück.

```
public String getDifinition(String word) {
    String difinition = null;
    Iterator<LanguagesSPI> dictionaryIterator = this.loader.iterator();
    while (difinition == null && dictionaryIterator.hasNext()) {
        difinition = dictionaryIterator.next().getDefinition(word);
    }
    return difinition;
}
```

Service Providers

- Implementierungen der Clients, und müssen die Schnittstelle “LanguagesSPI” implementieren.
-

```
@Override
public String getDefinition(String word) {
    if(this.words.containsKey(word))
        return words.get(word);
    else
        return null;
}
```

Lass uns praktisch die SPs an unseren Wörterbuch ankleben.

(Plug & Play)

Quellen

- https://docs.oracle.com/javase/tutorial/sound/SPI-intro.html
- https://www.journaldev.com/31602/java-spi-service-provider-interface-and-serviceloader
- https://www.baeldung.com/java-spi


THANK
YOU!