



ResolutionStrategy Interface

Interface for a resolution strategy.

Namespace

[CommerceExtension](#)

Usage

When you implement this interface, you can register your apex class just like an extension provider class. Your class can then conditionally decide how to handle each extension invocation. You can delegate to a specific extension provider, you can execute default domain logic, or you can execute no logic at all.

- [ResolutionStrategy Methods](#)
- [ResolutionStrategy Example Implementation](#)

ResolutionStrategy Methods

The following are methods for `ResolutionStrategy`.

- **`resolve()`**
Returns a resolution object, which indicates how the resolution strategy was resolved. The resolution indicates whether default logic, extension provider logic, or no logic is executed.

`resolve()`

Returns a resolution object, which indicates how the resolution strategy was resolved. The resolution indicates whether default logic, extension provider logic, or no logic is executed.

Signature

```
public CommerceExtension.Resolution resolve()
```

Return Value

Type: [CommerceExtension.Resolution](#)

Resolution object that indicates how the resolution strategy was resolved.

ResolutionStrategy Example Implementation

This is an example implementation of the `CommerceExtension.ResolutionStrategy` interface.

```
// This sample is for the situation when different tax behaviors need to be
// implemented for different locales.
//
// These tax behaviors can be -
// 1. ResolutionState - EXECUTE_DEFAULT (the default Salesforce Internal Tax Api).
```



```
// default Salesforce Internal Tax Api.
//
// An Extension Resolver is a custom apex class which selects different resolution
// states (EXECUTE_DEFAULT, EXECUTE_REGISTERED and OFF) for different locales
// to execute respective implementations (Extension Providers or the Default
// Salesforce Internal Tax Api).

// Your custom apex extension providers and the resolver must be registered with
// the tax extension point and then the resolver must be registered and mapped to
// the web store via appropriate setup.
//
// You can have as many Extension Providers registered as per your use case and
// select them in your resolver for different locales.
//
// Please follow the corresponding salesforce documentation on how to use locales.
// For more information related to that, please see the corresponding documentation.

// This must implement the commercestoretax.TaxService class in order to be
// processed by the tax service flow. It must also implement the
// CommerceExtension.ResolutionStrategy in order to work as a extension resolver
// and get the different locales and resolutions.
//
public class TaxServiceExtensionResolverSample
    extends commercestoretax.TaxService
    implements CommerceExtension.ResolutionStrategy {

    public CommerceExtension.Resolution resolve() {
        // The Sample Extension Provider registered with developer name as
        // 'tax_extension_provider_for_us' will be selected for execution for en_US locale
        if (CommerceExtension.ExtensionInfo.getLocaleString() == 'en_US') {
            return new CommerceExtension.Resolution('tax_extension_provider_for_us');
        }
        // The Sample Extension Provider registered with developer name as
        // 'tax_extension_provider_for_canada' will be selected for execution for en_CA locale
        if (CommerceExtension.ExtensionInfo.getLocaleString() == 'en_CA') {
            return new CommerceExtension.Resolution('tax_extension_provider_for_canada');
        }
        // The default Salesforce Internal Tax Api will return an empty response for Germany
        if (CommerceExtension.ExtensionInfo.getLocaleString() == 'de') {
            return new CommerceExtension.Resolution(
                CommerceExtension.ResolutionStates.OFF
            );
        }
        // The default Salesforce Internal Tax Api will be selected for execution for
        // all other locales than US, Canada and Germany
        return new CommerceExtension.Resolution();
    }
}
```

DID THIS ARTICLE SOLVE YOUR ISSUE?

Let us know so we can improve!

[Share your feedback](#)



[Commerce Cloud](#)
[Lightning Design System](#)
[Einstein](#)
[Quip](#)

[Trailhead](#)
[Sample Apps](#)
[Podcasts](#)
[AppExchange](#)

[Blog](#)
[Salesforce Admins](#)
[Salesforce Architects](#)

© Copyright 2025 Salesforce, Inc. [All rights reserved.](#) Various trademarks held by their respective owners. Salesforce, Inc.
Salesforce Tower, 415 Mission Street, 3rd Floor, San Francisco, CA 94105, United States

[Privacy Information](#) [Terms of Service](#) [Legal](#) [Use of Cookies](#) [Trust](#) [Cookie Preferences](#)

[Your Privacy Choices](#) [Responsible Disclosure](#) [Contact](#)