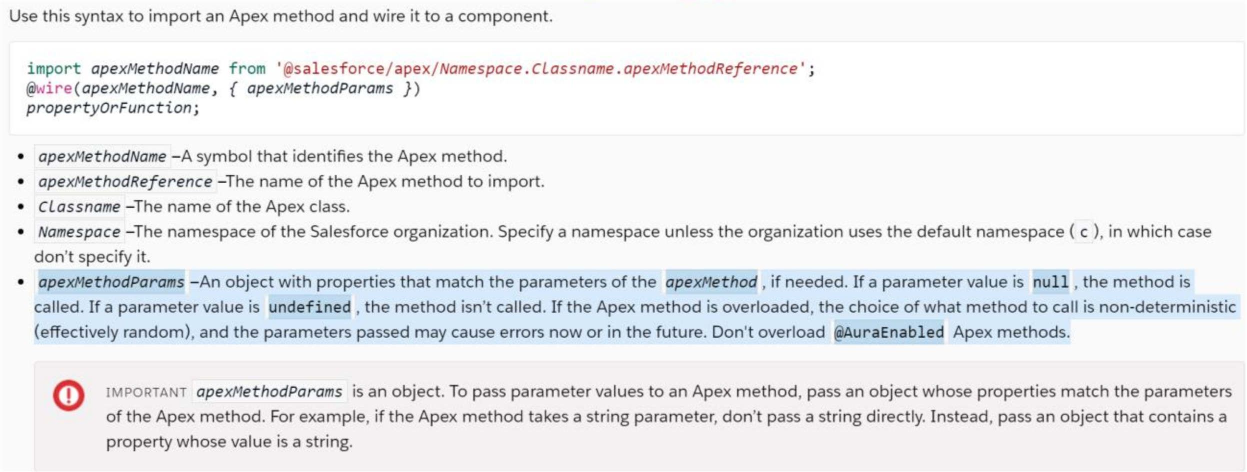
Cannot call overloaded methods in Apex using @Wire Decorator



Use this syntax to import an Apex method and wire it to a component.

```
import apexMethodName from '@salesforce/apex/Namespace.Classname.apexMethodReference';
@wire(apexMethodName, { apexMethodParams })
propertyOrFunction;
```

- `apexMethodName` –A symbol that identifies the Apex method.
- `apexMethodReference` –The name of the Apex method to import.
- `Classname` –The name of the Apex class.
- `Namespace` –The namespace of the Salesforce organization. Specify a namespace unless the organization uses the default namespace (`c`), in which case don't specify it.
- `apexMethodParams` –An object with properties that match the parameters of the `apexMethod`, if needed. If a parameter value is `null`, the method is called. If a parameter value is `undefined`, the method isn't called. If the Apex method is overloaded, the choice of what method to call is non-deterministic (effectively random), and the parameters passed may cause errors now or in the future. Don't overload `@AuraEnabled` Apex methods.

> IMPORTANT `apexMethodParams` is an object. To pass parameter values to an Apex method, pass an object whose properties match the parameters of the Apex method. For example, if the Apex method takes a string parameter, don't pass a string directly. Instead, pass an object that contains a property whose value is a string.