

Customizing AutoFixture Object Creation



Jason Roberts

.NET DEVELOPER

@robertsjason dontcodetired.com



Overview



Customize values generated by a Fixture

- Inject a specific instance for a type
- Register a custom creation algorithm
- “Freezing” values

Use AutoFixture’s Test Data Builder pattern implementation

Specimens and the AutoFixture pipeline

Customizations

- Example of prebuilt customization
- Create a custom specimen builder



Specimen

An individual value for a specific data type used as an example of that data type.

E.g. 1, 42, and 77 are specimens of int.

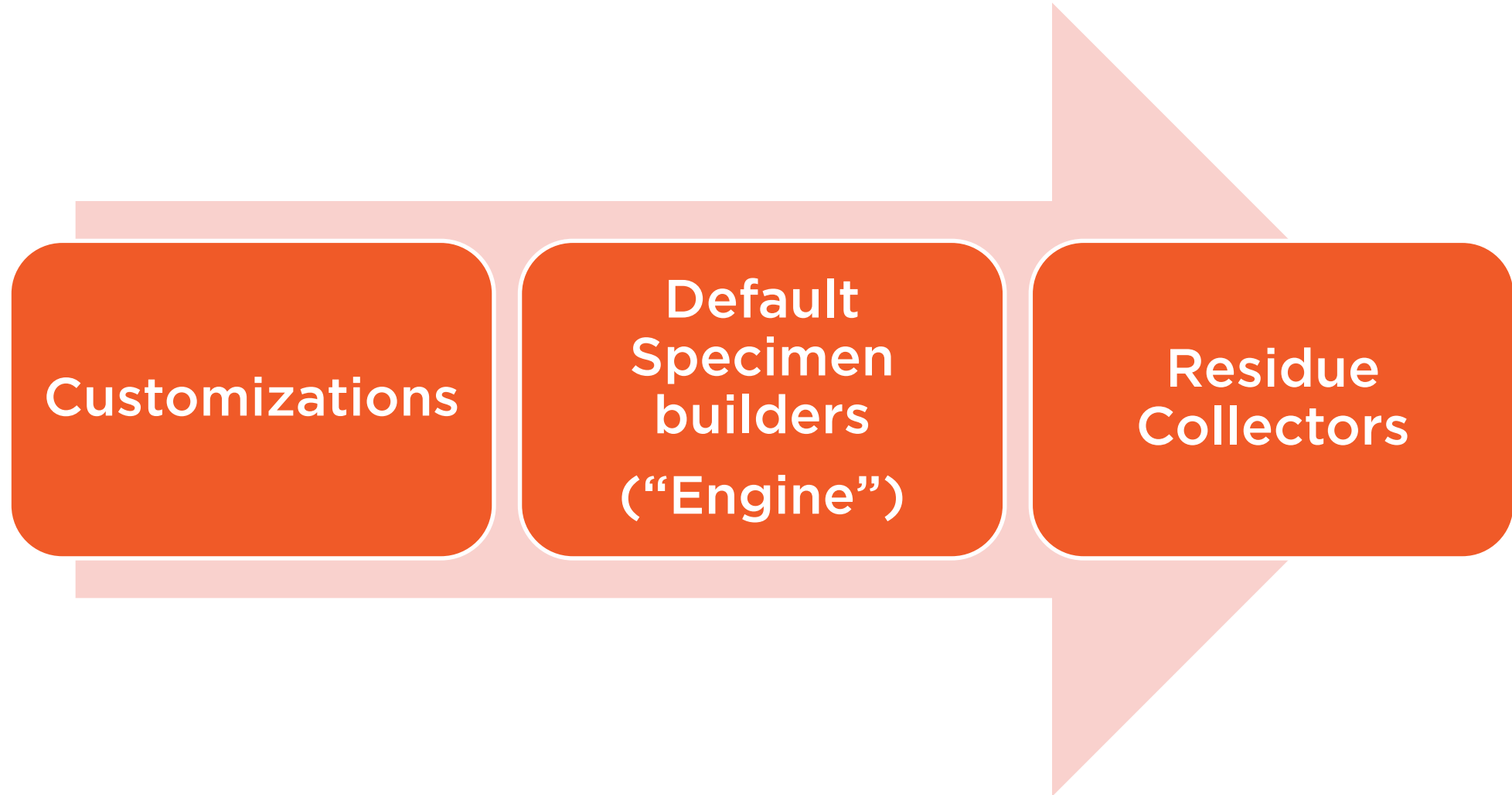


ISpecimenBuilder Interface

```
public interface ISpecimenBuilder
{
    object Create(object request,
                  ISpecimenContext context);
}
```



Fixture Pipeline Overview



Fixture Pipeline

Customizations

- Collection of ISpecimenBuilder
- “Override” default specimen builders
- Prebuilt & custom

Default Specimen builders

- “Engine”
- Preconfigured in a Fixture instance
- E.g. GuidGenerator specimen builder

Residue Collectors

- Lowest priority specimen builders
- Fall-back if no other matches
- “Last chance” before exception

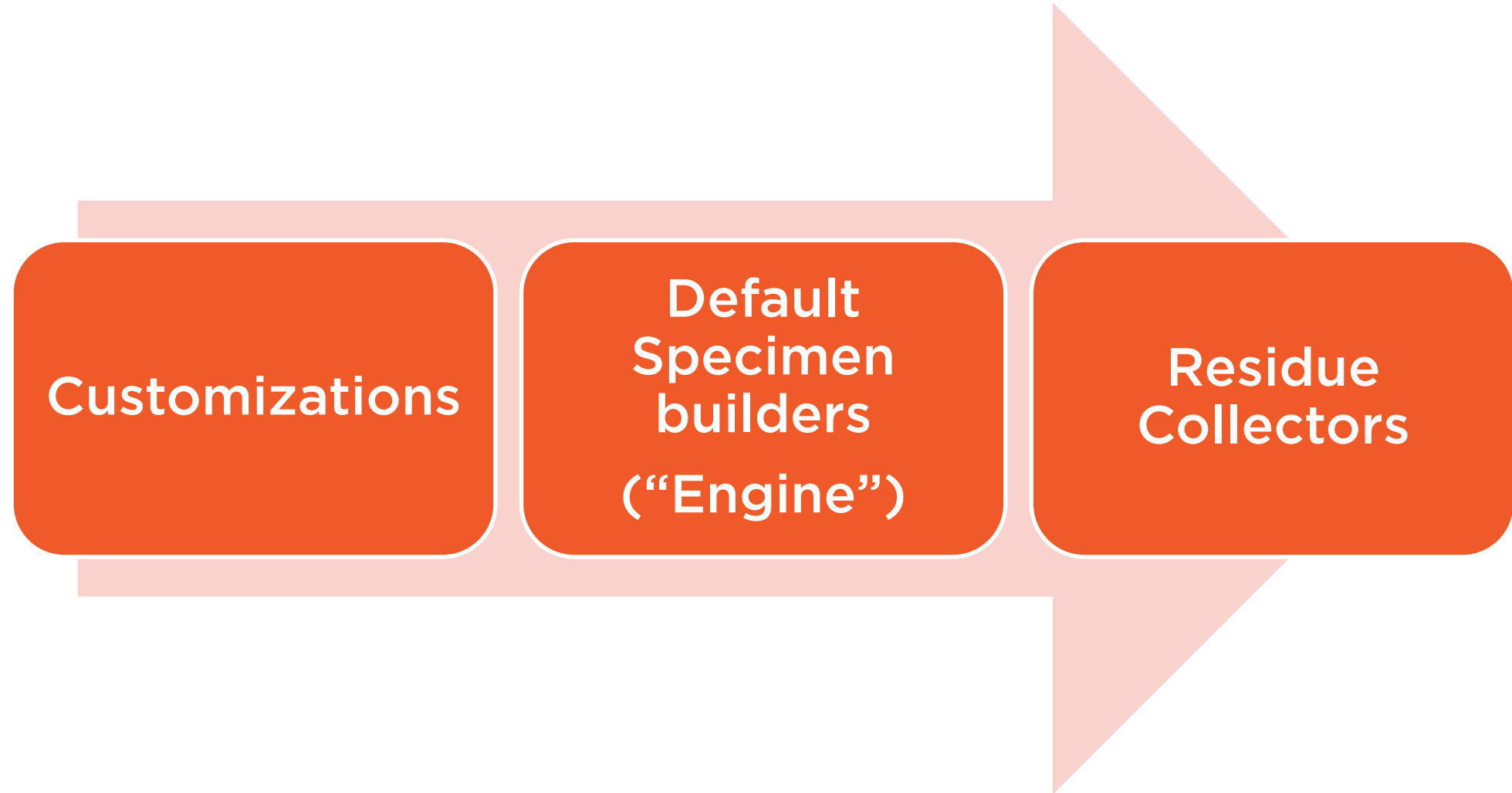


Essentially a request goes through a chain of ISpecimenBuilders until a specimen is generated.

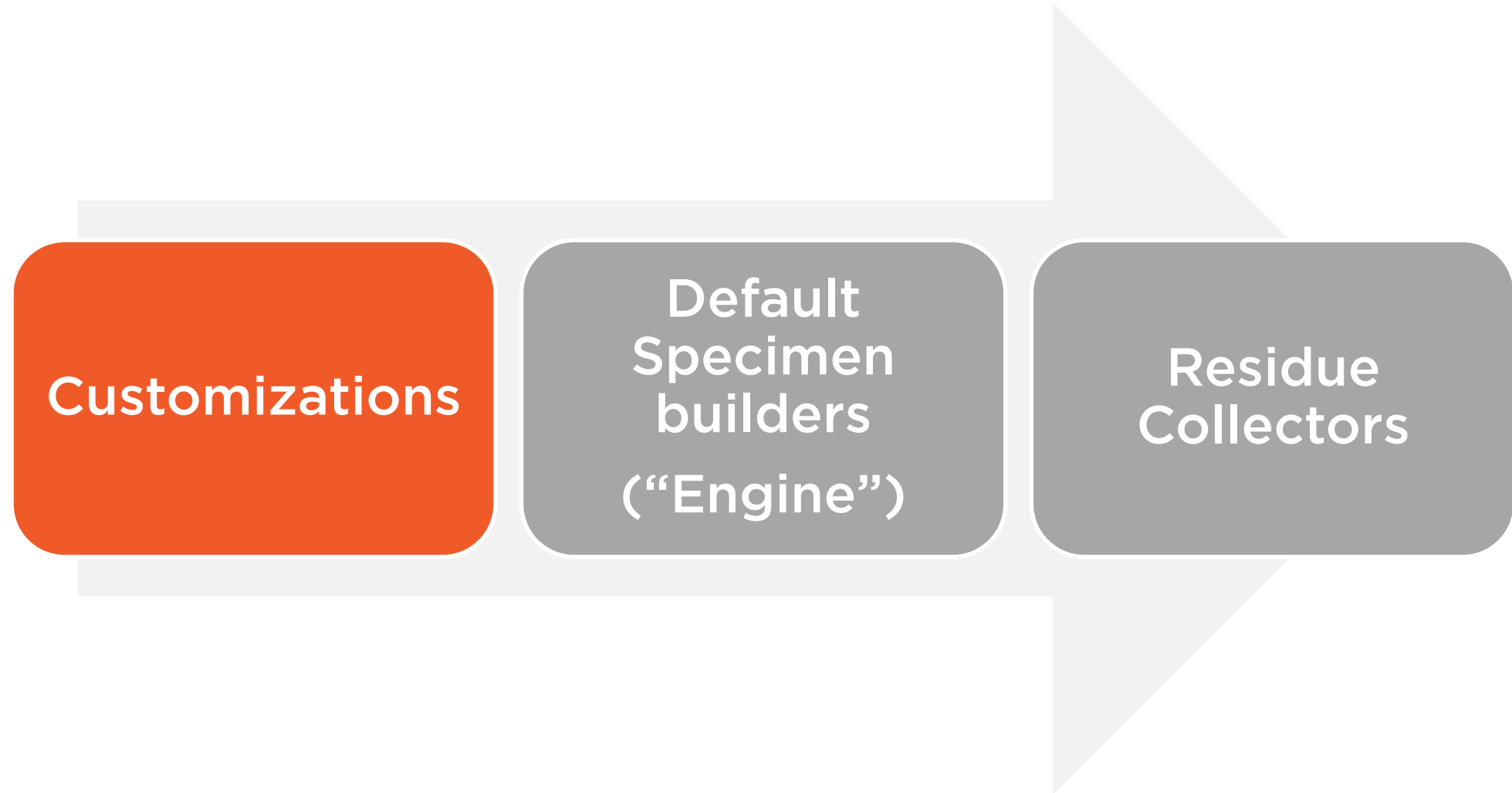
Request processing then stops and the specimen is returned to the test code.



Fixture Pipeline Overview



Fixture Pipeline Overview



Summary



Customize values generated by a Fixture

- `fixture.Inject("LHR")`
- `fixture.Register<string>(() => ...)`
- `fixture.Freeze<string>()`

`fixture.Build<FlightDetails>()`

Specimens and the AutoFixture pipeline

- Customizations
- Default specimen builders “Engine”
- Residue collectors

Customizations

- `CurrentDateTimeCustomization`
- `AirportCodeStringPropertyGenerator`



Up Next:
Writing Less Test Code and
Improving Test Maintenance

