



IEnumerable ISaveable IDontGetIt

Understanding .NET Interfaces

Presented by Jeremy Clark
www.jeremybytes.com



What are Interfaces?

Interfaces describe a group of related functions that can belong to any class or struct.

Public set of members:

- Properties
- Methods
- Events
- Indexers

Differences

Abstract Classes

- May contain implementation code
- A class may only descend from a single base class
- Members contain access modifiers
- May contain fields, properties, constructors, destructors, methods, events and indexers

Interfaces

- May not contain implementation code
- A class may implement any number of interfaces
- Members are automatically public
- May contain properties, methods, events, and indexers (not fields, constructors or destructors)

Various Data Sources

Microsoft SQL Server

MongoDB

CSV

SOAP Service

Oracle

WebAPI

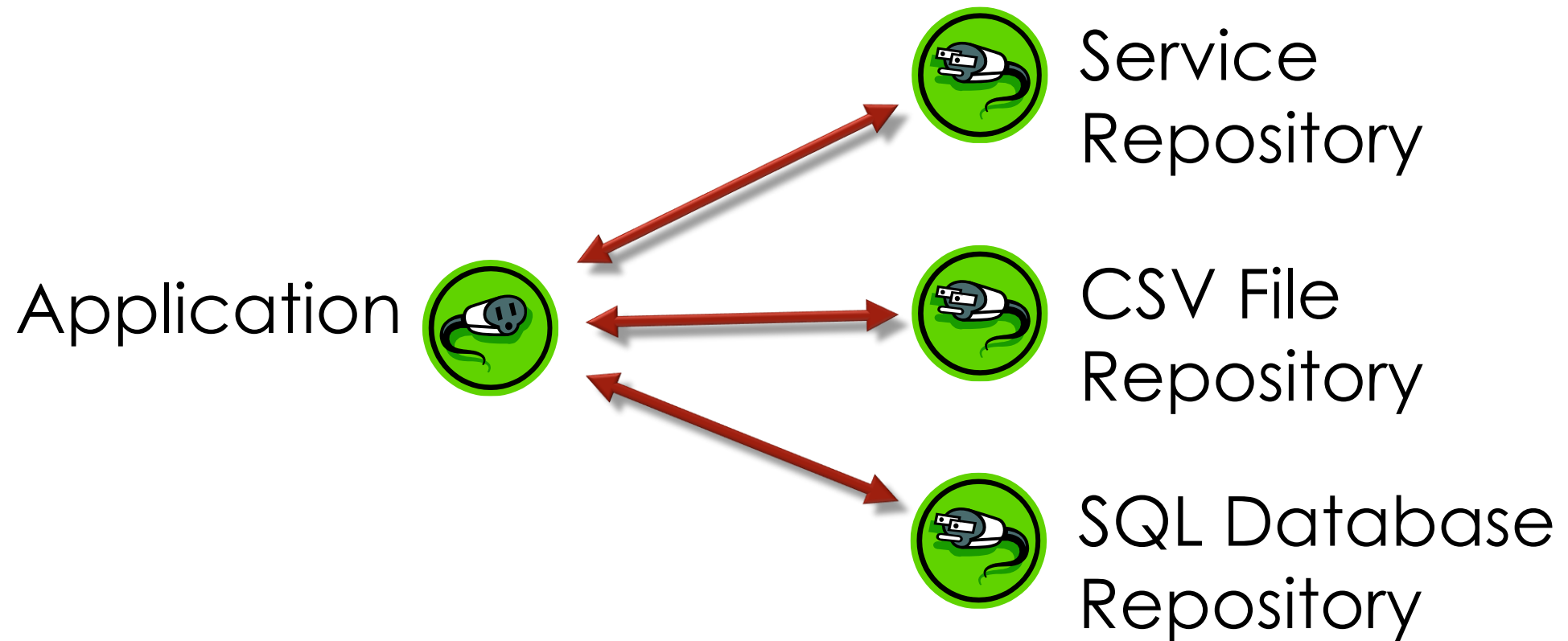
Amazon AWS

JSON

Microsoft Azure

Hadoop

Pluggable Repositories



Simple Repository

Create

Read

Update

Delete

Repository Interface

```
public interface IPersonRepository
{
C   void AddPerson(Person newPerson);
R   IEnumerable<Person> GetPeople();
    Person GetPerson(string lastName);
U   void UpdatePerson(string lastName,
    Person updatedPerson);
    void UpdatePeople(IEnumerable<Person>
    updatedPeople);
D   void DeletePerson(string lastName);
}
```




Differences

Abstract Classes

- May contain implementation code
- A class may only descend from a single base class
- Members contain access modifiers
- May contain fields, properties, constructors, destructors, methods, events and indexers

Interfaces

- May not contain implementation code
- A class may implement any number of interfaces
- Members are automatically public
- May contain properties, methods, events, and indexers (not fields, constructors or destructors)



Thank You!

Jeremy Clark

- <http://www.jeremybytes.com>
- jeremy@jeremybytes.com
- @jeremybytes