

Sun Mar 30 10:28 PM

academy.essentialdeveloper.com/ios-lead-essentials/447455/resources/8484976

◀ Previous Complete and Continue ▶ 

## iOS Lead Essentials • Part 1 Main Curriculum

Planning, Initial System Design and Requirements Analysis

- Are Singletons and Global Instances Damaging your System Design and Testability?
- Intro to Dependency Diagrams and Composition
- Intro to Modular Design
- BDD, TDD, Use Cases, Architecture and Modular Design
- Effectively Developing Swift Apps Before the Backend/Design Is Ready

### Writing/Reading Dependency Diagrams

There are different types of dependency you can represent in a dependency diagram using different annotations, lines, and arrows. Here are the main ones:

1. Solid line, empty head = "inherits from" / "is a".



A solid line with an empty head denotes that a class inherits from another class.

For example:

```
class MyViewController: UIViewController {}
```



The `MyViewController` class inherits from the `UIViewController` class, or the `MyViewController` "is a" subtype of `UIViewController`.

Screenshot 2024-0...51.42 AM

Sun Mar 30 10:28 PM

academy.essentialdeveloper.com/ios-lead-essentials/447455/resources/8484976

◀ Previous Complete and Continue ▶ 

## iOS Lead Essentials • Part 1 Main Curriculum

Planning, Initial System Design and Requirements Analysis

- Are Singletons and Global Instances Damaging your System Design and Testability?
- Intro to Dependency Diagrams and Composition
- Intro to Modular Design
- BDD, TDD, Use Cases, Architecture and Modular Design
- Effectively Developing Swift Apps Before the Backend/Design Is Ready

### 2. Dashed line, empty head = "conforms to" or "implements"

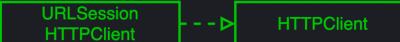


A dashed line with an empty head denotes that a component conforms/implements a protocol/abstract interface.

For example:

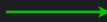
```
protocol HTTPClient {}
```

```
class URLSessionHTTPClient: HTTPClient {}
```



The `URLSessionHTTPClient` conforms to the `HTTPClient` protocol.

### 3. Solid line, filled head = "depends on" / "has a" (strong dependency)



A solid line with a filled head denotes a strong dependency between components.

Chrome File Edit View History Bookmarks Profiles Tab Window Help

RR vs CSK Live video stream | Intro to Dependency Diagrams | (46) WhatsApp | IRCTC Next Generation eTicket | Sun Mar 30 10:28PM

academy.essentialdeveloper.com/ios-lead-essentials/447455/resources/8484976

ESSENTIAL DEVELOPER

iOS Lead Essentials • Part 1 Main Curriculum

Planning, Initial System Design and Requirements Analysis

- Are Singletons and Global Instances Damaging your System Design and Testability?
- Intro to Dependency Diagrams and Composition
- Intro to Modular Design
- BDD, TDD, Use Cases, Architecture and Modular Design
- Effectively Developing Swift Apps Before the Backend/Design Is Ready

3. Solid line, filled head = "depends on" / "has a" (strong dependency)

A solid line with a filled head denotes a strong dependency.

When a type instance depends on another type instance to exist, it's considered a stronger dependency, such as Association, Aggregation, and Composition.

For example:

```
class RemoteFeedLoader {  
    private let client: HTTPClient  
  
    init(client: HTTPClient) {  
        self.client = client  
    }  
}
```

Screenshot 2024-0...51.42 AM

Chrome File Edit View History Bookmarks Profiles Tab Window Help

RR vs CSK Live video stream | Intro to Dependency Diagrams | (46) WhatsApp | IRCTC Next Generation eTicket | Sun Mar 30 10:28PM

academy.essentialdeveloper.com/ios-lead-essentials/447455/resources/8484976

ESSENTIAL DEVELOPER

iOS Lead Essentials • Part 1 Main Curriculum

Planning, Initial System Design and Requirements Analysis

- Are Singletons and Global Instances Damaging your System Design and Testability?
- Intro to Dependency Diagrams and Composition
- Intro to Modular Design
- BDD, TDD, Use Cases, Architecture and Modular Design
- Effectively Developing Swift Apps Before the Backend/Design Is Ready

You cannot instantiate a `RemoteFeedLoader` without an `HTTPClient` instance. The code wouldn't even compile. So that's a strong dependency.

The `RemoteFeedLoader` depends on an `HTTPClient` to exist.

4. Dashed line, filled head = "depends on" (weak dependency)

A dashed line with a filled head denotes a weak dependency.

It's important to note that a type can depend on and use another but still work without one.

For example:

```
class RemoteFeedLoader {  
    func load(with client: HTTPClient) {  
        client.doSomething()  
    }  
}
```

Screenshot 2024-0...51.42 AM

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Sun Mar 30 10:28PM

RR vs CSK Live video stream | Intro to Dependency Diagrams | (46) WhatsApp | IRCTC Next Generation eTicket

academy.essentialdeveloper.com/ios-lead-essentials/447455/resources/8484976

ESSENTIAL DEVELOPER

iOS Lead Essentials • Part 1 Main Curriculum

Search

Planning, Initial System Design and Requirements Analysis

- Are Singletons and Global Instances Damaging your System Design and Testability?
- Intro to Dependency Diagrams and Composition
- Intro to Modular Design
- BDD, TDD, Use Cases, Architecture and Modular Design
- Effectively Developing Swift Apps Before the Backend/Design Is Ready

class RemoteFeedLoader {  
 func load(with client: HTTPClient) {  
 client.doSomething()  
 }  
}

RemoteFeedLoader - -> HTTPClient

The `RemoteFeedLoader` has a source code dependency to the `HTTPClient` because it references and uses it. But it doesn't require an `HTTPClient` instance to exist.

You can create a `RemoteFeedLoader` without an `HTTPClient`.

That's considered a weaker dependency, but still a dependency!

The `RemoteFeedLoader` "uses an" `HTTPClient` dependency in the `load` method. But it doesn't "have" one. It must be provided as a parameter.

Screenshot  
2024-0...51:42 AM