

## Описание

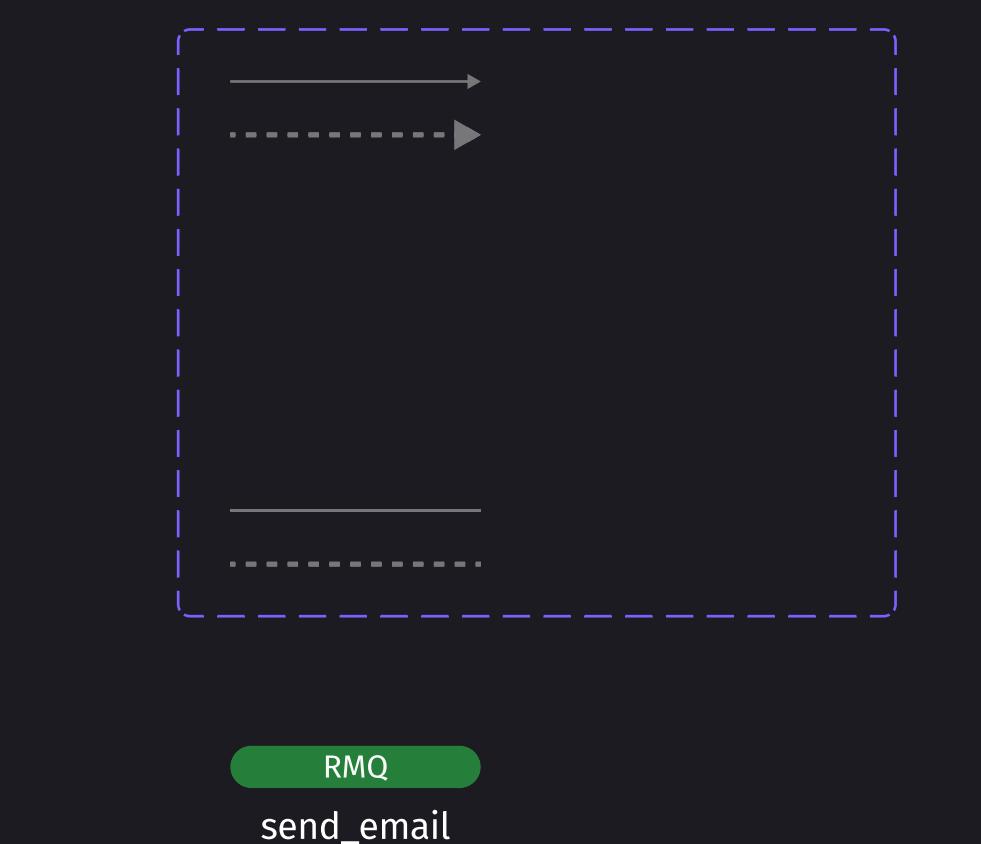
### Компонент

Это изолированные компоненты системы, которые вместе объединяются в модули. Обычно один компонент - один класс или файл где находится изолированная функциональность. В описании необходимо дать **название, тип и дополнительно описание**, если это необходимо. Варианты отображения:

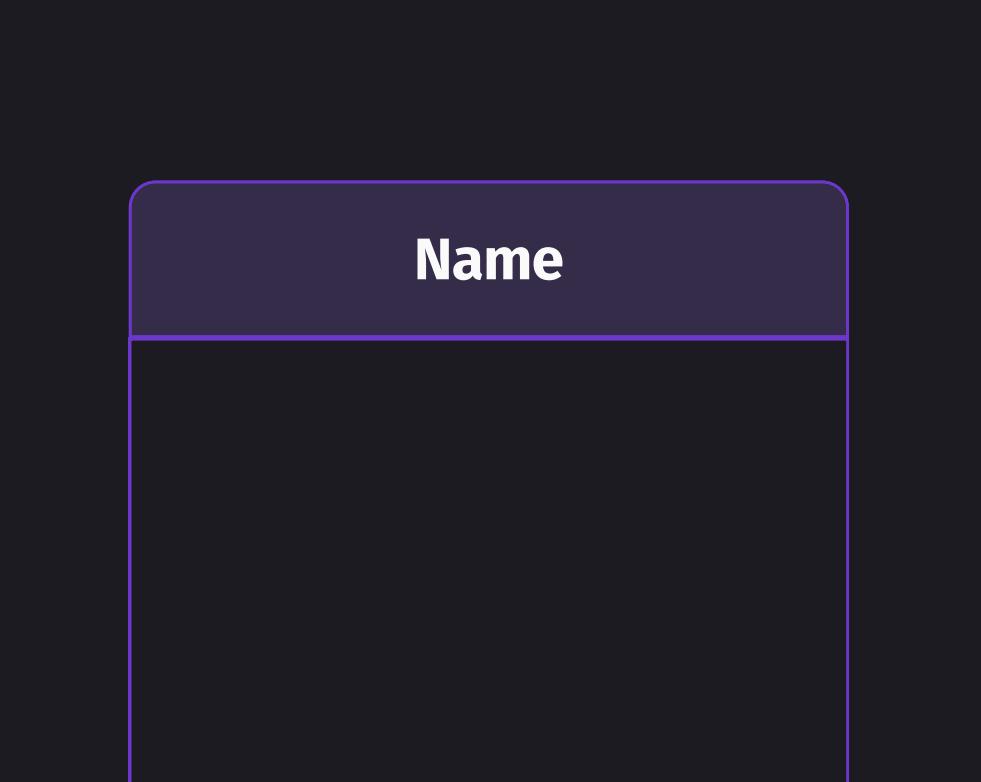


### Связи

Неразрывной стрелкой показываются связи между компонентами системы. При этом направление стрелки указывает направление вызова методов. Если компонент Users Controller вызывает методы Users Service:

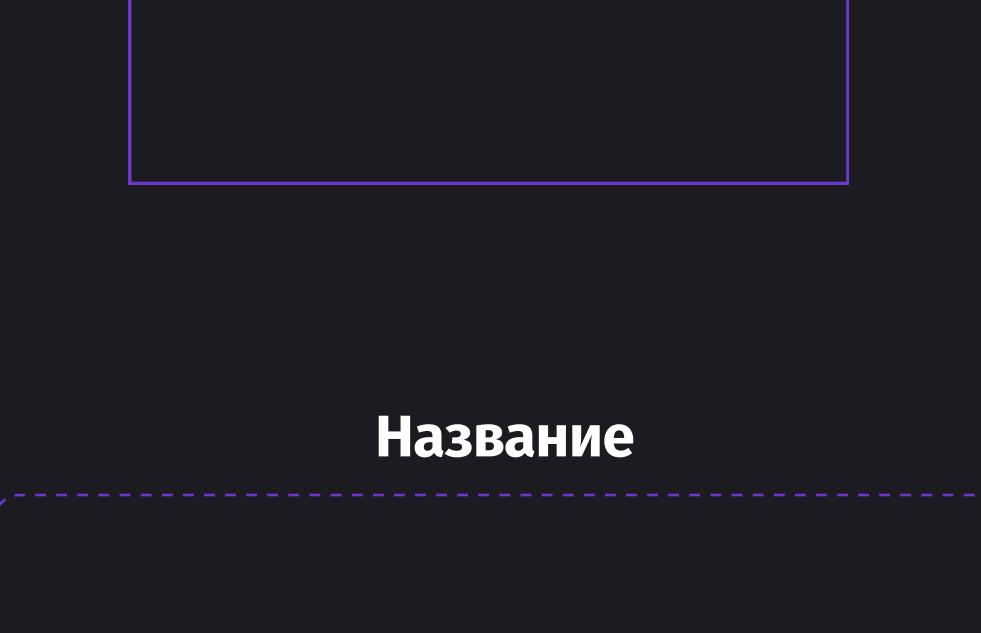


Пунктирной стрелкой обозначается входящее или исходящее сообщение / событие в систему из / в внешней системы. Это может быть API запросы, сообщения в другие части системы, вне описываемой системы или интеграции с внешними API. В описании необходимо дать название **команды** и её **тип и внешний сервис**, если необходимо. Каждый такая стрелка должна или исходить из компонента, или попадать в компонент. Примеры:



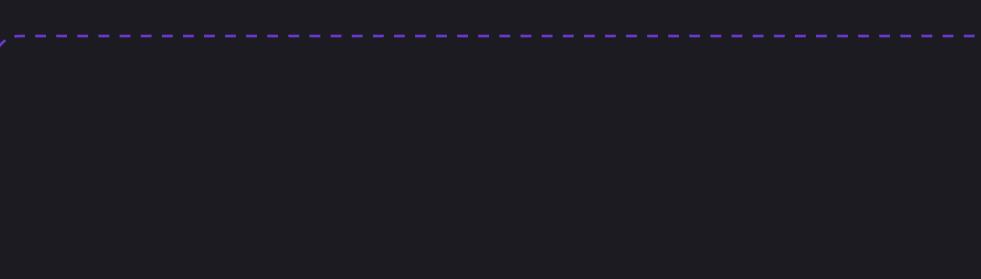
### Модуль

Содержат компоненты, позволяя объединить в себя функциональность одной отдельной области приложения, часто доменной. Каждый компонент может одновременно находиться только в одном модуле.

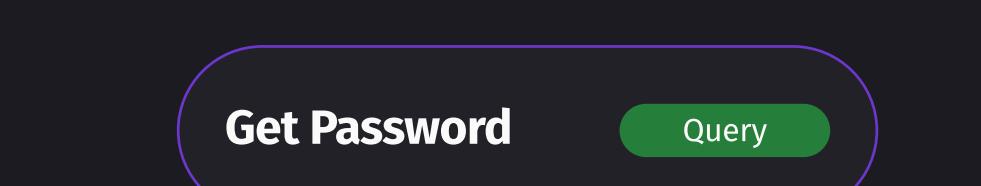


### Приложение

В случае если вам необходимо описать архитектуру, в которой участвует несколько приложений, можно использовать обёртку приложения. Пунктирными стрелками отмечается связь между ними. При этом каждое приложение в свою очередь может состоять из модулей.



На связях по аналогии с внешними связями можно указать **команду** и **протокол** взаимодействия. Над приложением указывается его название.

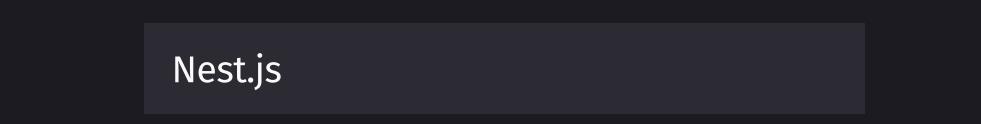


### Передаваемый объект

В случае если вам необходимо описать архитектуру, в которой участвует несколько приложений, можно использовать обёртку приложения. Пунктирными стрелками отмечается связь между ними. При этом каждое приложение в свою очередь может состоять из модулей.



На связях по аналогии с внешними связями можно указать **команду** и **протокол** взаимодействия. Над приложением указывается его название.



### Детализация

Для вашего приложения вы можете добавить список зависимостей, которые влияют на выбор архитектуры. Это делается с помощью компонента детализации, который является простым списком.

Так же если в рамках вашей архитектуры вам важно детально описать поля базы данных, или какой-либо объекта вы тоже можете воспользоваться детализацией.

## Библиотека

## Пример

