

Описание технической архитектуры

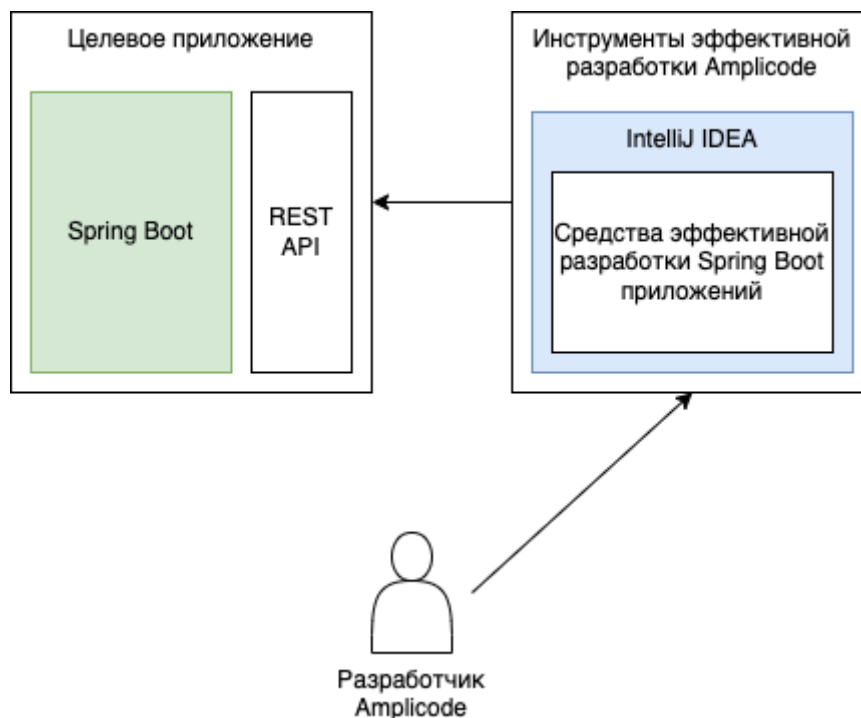
Программного обеспечения «Amplіcode»

Описание технической архитектуры программного обеспечения

Данный документ описывает базовые принципы и подходы к реализации различных частей разрабатываемого программного обеспечения Amplicode, а именно на инструментах эффективной разработки приложений.

Задача Amplicode сделать разработку современных бизнес приложений наиболее эффективной для профессиональных разработчиков, при этом сохраняя максимальную гибкость, не накладывая принципиальных ограничений ни на конечное приложение, ни на процесс разработки. Для достижения поставленной задачи эффективности и гибкости упор в Amplicode сделан именно на инструментарий, автоматизирующий создание типового кода.

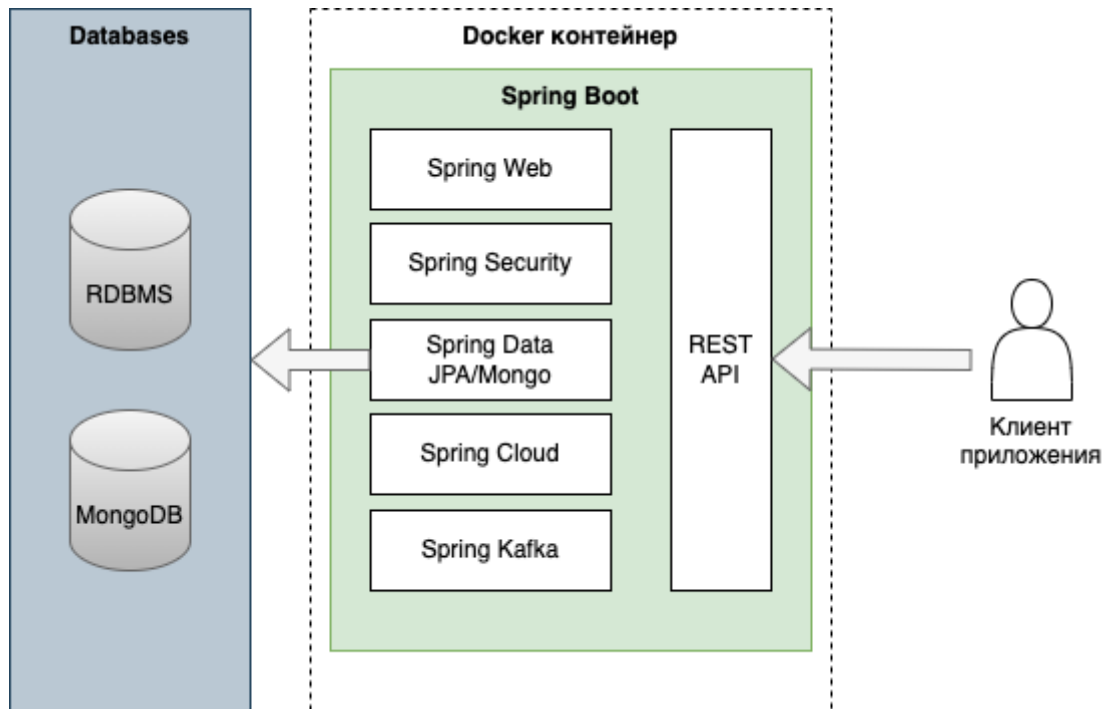
На диаграмме изображен процесс взаимодействия разработчика с Amplicode. Разработчик устанавливает программное обеспечение Amplicode в среду разработки IntelliJ IDEA, и используя инструментарий Amplicode, создает целевое бизнес приложение.



Целевое бизнес приложение строится на широкоиспользуемых, современных и проверенных технологиях. Выбор стека производился на основе множества исследований об использовании стека технологий, таких как *Developer Survey from stackoverflow* (<https://insights.stackoverflow.com/survey/2021>), *the state of developer ecosystem* (<https://www.jetbrains.com/lp/devecosystem-2021/>) от компании JetBrains и других отчетов. Также проводилось собственное анкетирование разработчиков, для более детального выяснения отношения к той или иной технологии.

В результате выбор сделан в пользу экосистем Spring. Spring широкоиспользуемый и проверенный фреймворк для разработки приложений на Java, который предлагает встроенные функции для управления зависимостями, конфигурации, безопасности и взаимодействия с базой данных. Модули Spring экосистемы: Spring Web, Spring Security, Spring Data JPA/Mongo, Spring Kafka, Spring Cloud и прочие модули.

Диаграмма отображает принципиальное устройство целевого приложения, разработанного на Amplicode:



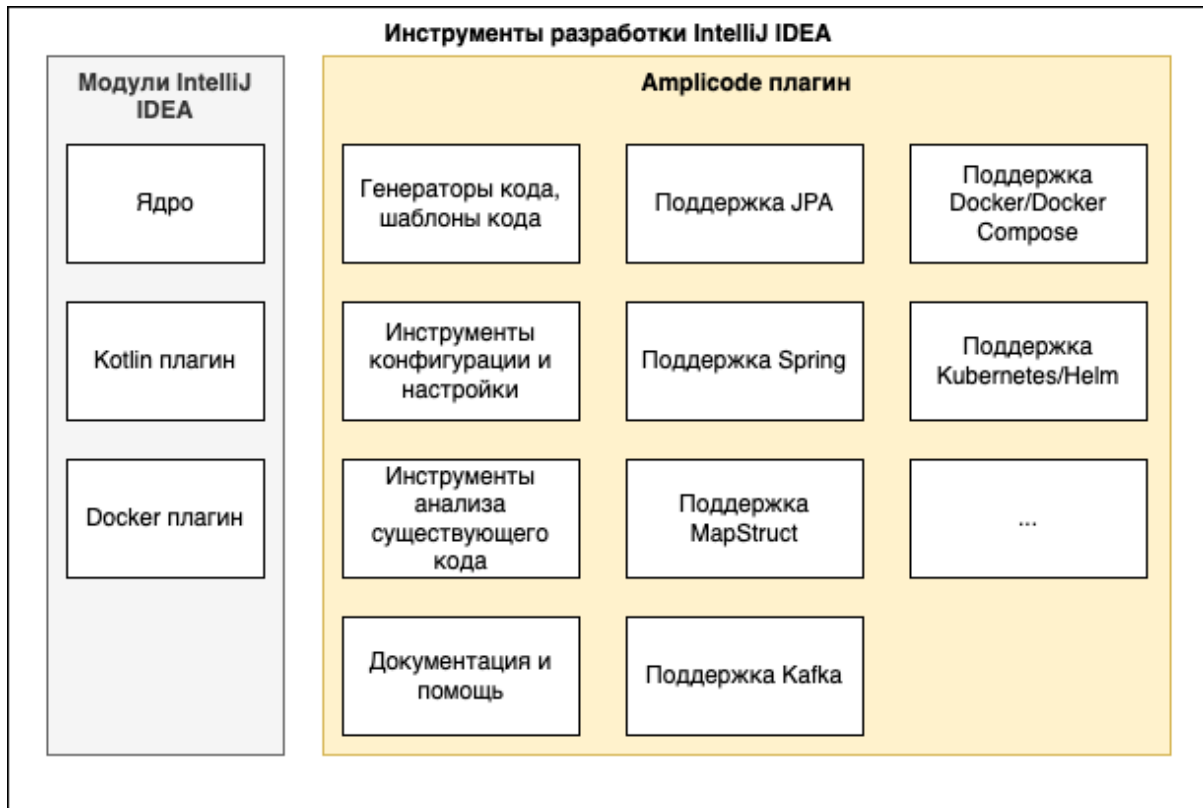
- **Spring Boot** используется в качестве основы для серверной части приложения. Это популярный фреймворк для разработки приложений на Java, который предлагает встроенные функции для управления зависимостями, конфигурации, безопасности и взаимодействия с базой данных.
- **Spring Data JPA/Mongo.** Приложение изначально рассчитана на работу с реляционными базами данных с использованием JPA Persistence API, но архитектура позволяет расширять поддержку и на другие типы хранилищ данных, например использую MongoDB.
- **Spring Web** позволяет **создавать REST API** для взаимодействия с клиентской частью. Это упрощает создание интерфейсов и интеграцию с другими системами. REST API обеспечивает простоту и гибкость.
- **Spring Security** используется для настройки ролевой модели и разграничений прав доступа, а так же для настройки различных схем аутентификации, например с использованием SSO.
- **Spring Kafka** упрощает взаимодействие с брокером сообщений Kafka, позволяя отправлять и обрабатывать сообщения.
- **Контейнеризация.** Приложения, созданные с помощью Amplicode, могут легко контейнеризоваться с помощью Docker. Это упрощает развертывание и управление приложениями в различных средах (например, в облаке или на локальных серверах).

Amplicode не накладывает широких ограничений на использование библиотек, совместимых с экосистемой Spring.

Плагин Amplicode для IntelliJ IDEA занимают центральную часть концепции. Именно плагин призван дать наибольший прирост эффективности при разработке бизнес

приложений. Плагин использовать механизмы IntelliJ для разбора и модификации кода, а также полагается на функционал уже существующих бесплатные плагины, такие как Kotlin, MapStruct, Docker.

Схема модулей, содержащихся в Amplicode плагине для IntelliJ IDEA, представлена на диаграмме ниже:



- **Ядро.** Плагин Amplicode взаимодействует с ядром IntelliJ IDEA через API, предоставляя доступ к различным функциям среды разработки, таким как редактор кода, управление файлами, поиску по индексам и т.д.. Плагин использует IntelliJ PSI (Program Structure Interface) для анализа и изменения структуры кода проекта.
- **Интеграция с существующими плагинами Kotlin/Docker.** Плагин Amplicode использует API существующих плагинов, либо расширяет функциональность плагинов.
- **Генераторы кода, шаблоны кода.** Плагин Amplicode использует генераторы кода для автоматизации создания различных элементов приложения, таких как сущности, репозитории, сервисы, REST API для взаимодействия с клиентской частью. Генераторы кода основаны на шаблонах (template files), которые содержат структуру и формат генерируемого кода.
- **Инструменты конфигурации и настройки.** Плагин Amplicode включает в себя средства для настройки проекта, такие как управление зависимостями Maven/Gradle, настройка баз данных, параметры развертывания и другие конфигурации, связанные с проектом.

- **Spring Kafka** упрощает взаимодействие с брокером сообщений Kafka, позволяя отправлять и обрабатывать сообщения.
- **Контейнеризация.** Приложения, созданные с помощью Amplicode, могут легко контейнеризоваться с помощью Docker. Это упрощает развертывание и управление приложениями в различных средах (например, в облаке или на локальных серверах).

Amplicode позволяет работать как в opensource версии IntelliJ Community Edition, так и в платной версии IntelliJ Ultimate.