Использование ИИ-агентов через MCP

Редакция от 19.10.2026 · docs.kb-agava.ru

Содержание

1 Условия использования 2
2 Настройка MCP интеграции 2
  2.1 Группа «MCP-сервер для ИИ-агентов» 2
  2.2 Разрешение ссылок на элементы скриптов 3
  2.3 Группа «Инструкции для агента в папке проекта» 3
  2.4 Адрес и порт подключения 4
3 Подключение агента 4
  3.1 Общий порядок 4
  3.2 Варианты, представленные в окне подключения 4
  3.3 Проверка выбранного проекта 5
4 Подготовка инструкций AGENTS.md 5
5 Доступные операции 6
  5.1 Редактирование программных текстов 7
  5.2 Предварительный просмотр графики 7
6 Порядок выполнения задания 7
7 Проверка в локальной среде исполнения 8
8 Сохранение и отмена изменений 8
9 Примеры заданий 9
  9.1 Изучение и проверка проекта 9
  9.2 Подготовка окна 9
  9.3 Проверка программной логики 9
10 Типовые затруднения 9

Универсальная среда разработки Agava предоставляет средства взаимодействия с внешними ИИ-агентами через **MCP (Model Context Protocol)**. Подключённый агент может изучать структуру открытого проекта, создавать и изменять узлы, редактировать программные тексты, проверять проект и исследовать его выполнение в локальной среде исполнения.

Пользователь формулирует задание в интерфейсе выбранного агента. Для выполнения задания агент обращается к инструментам DevEnv и получает сведения о проекте и результатах операций. Изменения доступны для просмотра в среде разработки.

Настройка модели, учётной записи и самого агента выполняется средствами соответствующего клиентского приложения. Раздел **«ИИ-агенты (MCP)»** окна "Настройки" предназначен для управления подключением к проекту и подготовки проектных инструкций.

1 Условия использования

Для начала работы необходимы:

- Универсальная среда разработки Agava с открытым проектом;
- установленный ИИ-агент с поддержкой MCP через HTTP либо с возможностью использования переходника для stdio;
- включённый MCP-сервер в среде разработки Agava;
- параметры подключения к среде разработки Agava, внесённые в конфигурацию агента.

MCP-сервер принимает подключения только по локальному адресу **127.0.0.1**. Подключение рассчитано на работу агента на том же компьютере, где запущена среда разработки. Стандартный адрес сервера:

```
http://127.0.0.1:8765/mcp
```

Универсальную среду разработки Agava следует запускать до агента. Рабочей папкой агента рекомендуется назначать папку проекта, открытого в среде разработки.

Объектом операций является проект, открытый в среде разработки. Выбор другой рабочей папки в клиенте агента сам по себе не переключает проект в среде разработки.

2 Настройка MCP интеграции

Для открытия параметров MCP интеграции необходимо выбрать в главном меню команду **«Параметры → Настройки → ИИ-агенты (MCP)»**.

2.1 Группа «MCP-сервер для ИИ-агентов»

Элемент	Назначение
«Включить MCP-сервер (Claude Code, Codex, Cursor и другие агенты)»	Разрешает подключение агентов к среде разработки. По умолчанию включён. Переключение непосредственно запускает или останавливает сервер.

«Разрешить агентам ссылки на глобальные переменные и функции скриптов»	Разрешает агенту назначать в свойствах узлов ссылки на глобальные переменные, функции и методы скриптов. По умолчанию выключен.
«Состояние»	Отображает результат запуска сервера, адрес подключения либо причину недоступности.
«Подключить агента...»	Открывает окно с параметрами подключения для различных клиентов.

В строке состояния могут отображаться следующие сведения:

Состояние	Значение
«работает, …»	Сервер запущен; далее приведён его фактический адрес.
«выключен в настройках»	Флажок включения сервера снят.
«не запущен: …»	Сервер не удалось запустить либо запуск отключён переменной окружения. После двоеточия указана причина.

Для сохранения параметров необходимо подтвердить окно настроек кнопкой **OK**. При отмене восстанавливаются ранее сохранённые параметры. При этом переключение флажка включения сервера действует уже во время работы с окном.

Остановка MCP-сервера прекращает доступ через это подключение. Выполненные ранее изменения проекта сохраняются в текущем состоянии среды разработки; остановка сервера их не отменяет.

2.2 Разрешение ссылок на элементы скриптов

По умолчанию при настройке источников данных и других свойств-ссылок агенту доступны подходящие узлы, сигналы и свойства узлов. Назначение ссылок на глобальные переменные, функции и методы скриптов отклоняется.

Если такая связь предусмотрена проектом, пользователь может установить флажок **«Разрешить агентам ссылки на глобальные переменные и функции скриптов»**. При этом для каждого свойства сохраняются обычные требования к допустимому виду и типу источника.

Параметр относится к ссылкам, назначаемым в свойствах проекта. Он не устанавливает запрет на использование глобальных переменных и вызов функций внутри программного текста. Снятие флажка не удаляет уже существующие связи проекта.

2.3 Группа «Инструкции для агента в папке проекта»

Группа предназначена для подготовки файла проектных соглашений **AGENTS.md**. Кнопка **«Создать AGENTS.md в папке открытого проекта»** размещает шаблон инструкций рядом с файлом проекта `.asp` и открывает его в приложении, связанном с файлами Markdown.

Создание доступно для открытого проекта формата `.asp`, имеющего путь к файлу. Если такой проект не открыт, среда разработки выводит соответствующее сообщение.

Подробный порядок подготовки инструкций приведён в разделе 5.

2.4 Адрес и порт подключения

Отдельного поля ввода порта на странице настроек нет. Стандартный порт **8765** может быть изменён переменной окружения **AGAVA_DEVENV_MCP_PORT**, заданной для процесса DevEnv.

Значение переменной	Результат
Переменная не задана	Используется порт 8765.
Целое число от 1 до 65535	Используется указанный порт.
`0`	Запуск MCP-сервера отключён независимо от флажка в настройках.
Недопустимое значение	Сервер не запускается; в состоянии отображается сообщение об ошибке.

После изменения окружения следует перезапустить среду разработки и использовать адрес, указанный в его строке состояния. Если адрес изменён, конфигурацию агента необходимо обновить.

3 Подключение агента

3.1 Общий порядок

1. Запустить универсальную среду разработки Agava и открыть требуемый проект.
2. вызвать команду меню **«Параметры → Настройки → ИИ-агенты (MCP)»**.
3. Убедиться, что сервер включён и строка состояния содержит **«работает»** и адрес подключения.
4. Нажать кнопку **«Подключить агента...»**.
5. В окне **«Подключить ИИ-агента»** выбрать блок, соответствующий используемому клиенту, и нажать **«Копировать»**.
6. Применить скопированную команду или фрагмент конфигурации в клиенте агента.
7. Подтвердить и закрыть окно настроек.
8. Запустить либо переподключить агент в папке проекта и проверить доступность MCP-сервера.

Окно подключения содержит готовые параметры с фактическим адресом сервера. Оно не устанавливает клиент, не изменяет его конфигурацию автоматически и не запускает сеанс агента.

Подключение обычно настраивается однократно для выбранного клиента на данном компьютере. Повторная настройка требуется при изменении адреса или конфигурации клиента.

3.2 Варианты, представленные в окне подключения

Блок окна	Предоставляемые параметры
Claude Code	Команда регистрации HTTP-сервера `devenv`, выполняемая в терминале.
Codex	Фрагмент для файла `~/.codex/config.toml` с разделом `[mcp_servers.devenv]` и адресом сервера.
Cursor и другие агенты с JSON-конфигурацией	JSON-фрагмент с записью `devenv` в разделе `mcpServers`.

Агенты, которые умеют только stdio	Команда запуска переходника `npx -y mcp-remote` с адресом DevEnv. Для этого варианта требуется Node.js.

Если конфигурационный файл клиента уже содержит другие серверы, запись следует добавить к существующей конфигурации с сохранением её структуры.

Адрес `/mcp` предназначен для MCP-клиента. Открытие этого адреса в обычном браузере не является проверкой работоспособности подключения и не отображает пользовательскую веб-страницу.

3.3 Проверка выбранного проекта

Первое задание агенту должно включать получение сведений о проекте инструментом **`project_info`** с передачей его текущей рабочей папки в параметре **`workdir`**.

Среда разработки возвращает сведения об открытом проекте и сообщает о соответствии рабочей папки. При несовпадении выводится предупреждение: дальнейшие операции будут применены к проекту, открытому в среде разработки. Предупреждение не переключает проект и не блокирует последующие изменения.

До начала редактирования следует установить, какой проект требуется изменить. При необходимости агент может открыть другой файл `.asp` инструментом `open_project`. Если закрытие текущего проекта требует решения о сохранении изменений, соответствующий диалог отображается пользователю.

4 Подготовка инструкций AGENTS.md

Файл **AGENTS.md** содержит общие правила работы агента с конкретным проектом. В подготовленный шаблон файла включены сведения о подключении, проверке открытого проекта, изменении узлов через MCP, ограничениях ссылок и проверке результатов.

Пользователю следует дополнить шаблон следующими сведениями:

- правила именования узлов и программных элементов;
- структура станций, групп, сигналов и приложений;
- принятые адреса и обозначения оборудования;
- требования к размерам окон, цветам, шрифтам и размещению элементов;
- условия выполнения задания, проверки и сохранения результатов.

Порядок подготовки файла:

1. Открыть требуемый проект `.asp` в среде разработки.
2. На странице **«ИИ-агенты (MCP)»** нажать **«Создать AGENTS.md в папке открытого проекта»**.
3. Дополнить открывшийся файл соглашениями проекта и сохранить его.
4. Убедиться, что выбранный агент использует этот файл инструкций. Способ подключения инструкций определяется клиентом.

Если **AGENTS.md** уже существует, DevEnv предлагает заменить его шаблоном. Ответ **«Нет»**, выбранный по умолчанию, сохраняет содержимое и открывает существующий файл. Ответ **«Да»** заменяет файл шаблоном, поэтому ранее внесённые соглашения необходимо учитывать перед подтверждением.

При записи шаблона рядом также создаётся **CLAUDE.md** со строкой `@AGENTS.md`, если файла CLAUDE.md ещё нет. Существующий CLAUDE.md не изменяется. Если AGENTS.md был только открыт без замены, создание сопутствующего файла на этом шаге не выполняется.

AGENTS.md задаёт инструкции для агента. Ограничение назначения ссылок на элементы скриптов определяется флажком DevEnv; изменение текста инструкций само по себе не изменяет это разрешение. После изменения разрешения соответствующий пункт шаблона следует привести в соответствие с принятыми правилами проекта.

5 Доступные операции

Перечень инструментов агент получает от среды разработки при подключении. Пользователю достаточно задавать требуемый результат; указание имён инструментов полезно для уточнения порядка работы и проверки.

Задача пользователя	**Возможности и основные инструменты**
Изучить проект	Получение сведений о проекте (`project_info`), дерева (`get_tree`), поиск узлов (`find_nodes`), п свойств (`describe_node`, `get_value`).
Выбрать допустимый тип узла или источник данных	Получение описаний типов, допустимых дочерних узлов и целей ссылок (`describe_type`, `list` `list_link_targets`).
Изменить структуру проекта	Создание, добавление, перемещение и удаление узлов (`create_node`, `add_nodes`, `move_node`, `de`
Настроить свойства и связи	Изменение значений и сложных свойств (`set_values`, `set_properties`).
Изменить программные тексты AS и ST	Чтение текста, его замена и проверка (`get_value`, `set_script`, `check_script`).
Подготовить окно или композицию	Создание и настройка графических узлов, просмотр расположения виджетов (`list_widgets`) (`render_node`).
Проверить проект	Получение общей диагностики и результатов компиляции (`check_all`); отдельная проверка п (`validate_project`).
Работать с документами	Просмотр и изменение узлов коллекции документов, получение таблиц данных (`document_da` макетом (`document_template`), формирование PDF (`document_export`).
Проверить выполнение	Запуск, остановка и просмотр состояния локальной среды исполнения (`runtime_start`, `runti` `runtime_status`).
Исследовать поведение программы и интерфейса	Чтение и запись значений, нажатие виджетов, получение снимков, диалогов и сообщений (`r` `runtime_write`, `runtime_widgets`, `runtime_click`, `runtime_screenshot`, `runtime_dialog`, `runtime_l`
Управлять результатом редактирования	Пакет изменений (`batch`), отмена последнего действия (`undo_last`), сохранение проекта (`sav`

В коллекции документов используются те же общие операции с узлами, но выбирается коллекция **`documents`**. Идентификатор узла следует рассматривать вместе с его коллекцией.

5.1 Редактирование программных текстов

При замене текста инструментом `set_script` выполняется компиляция и возвращаются диагностические сообщения. Если текст открыт в текстовом редакторе среды разработки, содержимое вкладки также обновляется.

Несохранённые ручные изменения этой вкладки заменяются текстом, переданным агентом. До поручения такой операции следует согласовать актуальную редакцию текста.

Имя программы или функции, аргументы и тип результата следует изменять через соответствующие свойства узла. Строка прототипа в программном тексте должна соответствовать этим свойствам.

5.2 Предварительный просмотр графики

Инструмент `render_node` показывает окно, композицию, слой или содержащий виджет экран в представлении редактора. При включении подписей агент может сопоставить элементы изображения с идентификаторами узлов. Значения отображаются в состоянии разработки.

Для оценки фактически работающего интерфейса используется `runtime_screenshot` после запуска среды исполнения. Предварительное изображение редактора и снимок работающей программы применяются на разных этапах проверки.

6 Порядок выполнения задания

1. **Определить цель.** Указать требуемый результат, узлы и ограничения изменения проекта.
2. **Проверить контекст.** Поручить агенту вызвать `project_info` с рабочей папкой и подтвердить соответствие открытого проекта.
3. **Изучить исходное состояние.** Получить структуру, свойства, существующие связи и программные тексты, относящиеся к заданию.
4. **Внести изменения через MCP.** Для связанной последовательности изменений при необходимости использовать `batch`.
5. **Выполнить проверку.** Запросить `check_all`, рассмотреть ошибки, предупреждения и замечания. Для отображения замечаний в ответе можно указать `min_severity: "notice"`.
6. **Проверить графику.** Для изменённых окон и композиций получить предварительные изображения и проверить расположение виджетов.
7. **Проверить выполнение, если оно входит в задание.** Запустить локальную среду исполнения, выполнить заданные воздействия, сравнить полученные значения и поведение с ожидаемыми.
8. **Оценить результат.** Просмотреть изменённые узлы и полученную диагностику в DevEnv.
9. **Сохранить проект.** Выполнить сохранение средствами среды разработки, либо поручить агенту вызвать `save_project`.

Проверка `check_all` сообщает о критических проблемах, ошибках, предупреждениях и замечаниях, а также выполняет компиляцию скриптов приложений. Наличие критических проблем или ошибок приводит к отрицательному результату проверки. Успешная проверка не заменяет проверку поведения при выполнении.

Чтобы изменения, вносимые агентом в проект, согласованно отображались в среде разработки, агенту следует поручать редактирование проекта инструментами MCP. Прямые правки файлов `.asp`, каталога узлов и других файлов открытого проекта могут быть перезаписаны при последующем сохранении проекта в среде разработки.

7 Проверка в локальной среде исполнения

Инструмент **`runtime_start`** формирует исполняемый файл `.bin` из текущего состояния проекта, запускает локальный **rtu.exe** без отладки и подключает к нему отладчик среды разработки. При этом сохранение исходного проекта `.asp` не выполняется.

Запуск через `runtime_start` останавливает уже запущенные процессы rtu.exe на компьютере. Инструмент `runtime_stop` также останавливает все процессы rtu.exe на этом компьютере. Это необходимо учитывать, если одновременно выполняются другие локальные проекты.

После запуска агент может:

- получить значения источников, сигналов и свойств узлов;
- выполнить несколько измерений с заданным интервалом для проверки изменений во времени;
- записать значение либо нажать виджет для проверки реакции программы;
- получить перечень виджетов и снимок работающего интерфейса;
- прочитать сообщения и сведения об открытых диалогах среды исполнения.

Запись значений и нажатие виджетов воздействуют на работающую программу. Сценарий проверки должен определять допустимые воздействия и ожидаемый результат.

Изменения, внесённые в проект после запуска RTU, начинают использоваться в этой проверке только после повторного вызова `runtime_start`. Обновление изображения редактора не означает обновления исполняемого проекта.

В конце проверки среду исполнения можно остановить инструментом `runtime_stop`. Вывод запущенного из DevEnv RTU остаётся доступным через `runtime_logs` до следующего запуска.

8 Сохранение и отмена изменений

Операции редактирования изменяют открытый проект. Для записи текущего состояния на диск предусмотрен `save_project`. Запуск RTU и успешное завершение проверки не заменяют сохранение проекта.

Изменения узлов, поддерживающие отмену, включаются в историю изменений проекта. Инструмент **`undo_last`** отменяет последнее действие в общей истории дерева проекта. Последним может оказаться как действие агента, так и ручное действие пользователя; перед отменой следует учитывать порядок операций.

Инструмент **`batch`** объединяет допустимые операции редактирования в одну запись отмены. По умолчанию при ошибке выполненные изменения пакета откатываются. Операции сохранения и переключения проекта, удаления узлов, запуска и остановки RTU, воздействия на RTU и записи файлов документов выполняются отдельно от такого пакета.

Отмена изменений узлов не является отменой всех внешних результатов работы агента. В частности:

- `create_project` сразу создаёт и сохраняет новый файл проекта;
- запись макета инструментом `document_template` сразу изменяет файл макета; отмена свойства пути не восстанавливает его содержимое;
- `document_export` сразу создаёт PDF; файл не удаляется отменой изменений дерева;
- запись значений и нажатия в работающей среде исполнения не отменяются через `undo_last`.

Для замены существующего макета или PDF инструменты документов требуют явного разрешения перезаписи. Требование сохранить прежние файлы следует включать в задание агенту.

9 Примеры заданий

9.1 Изучение и проверка проекта

```
Подключись к DevEnv через MCP. Сначала вызови project_info с текущей рабочей папкой и проверь соответствие открытого проекта. Изучи структуру приложения и задач. Выполни проверку проекта с включением замечаний и представь список проблем с указанием узлов. Изменять проект не требуется.
```

9.2 Подготовка окна

```
В открытом проекте подготовь окно отображения выбранного сигнала температуры. Используй существующий сигнал и допустимые цели ссылок. Соблюдай размеры, цвета и правила именования из AGENTS.md. После изменения выполни проверку проекта и покажи предварительное изображение окна. Сохрани проект после успешной проверки.
```

9.3 Проверка программной логики

```
Проверь выбранную программу на заданных входных значениях. Сначала изучи её текст и связи, затем выполни проверку проекта. Запусти локальный RTU, выполни согласованный сценарий изменения входов и получи выходные значения. Для проверки задержки используй серию измерений. Приведи ожидаемые и фактические результаты, после проверки останови RTU.
```

В последнем примере запуск предполагает остановку ранее запущенных локальных процессов rtu.exe, как указано в разделе 8.

10 Типовые затруднения

Ситуация	Действия пользователя
Агент не видит MCP-сервер	Проверить запуск среды разработки, флажок включения, строку состояния и применение конфигурации в клиенте. После изменения конфигурации переподключить агент.
Порт занят	Проверить наличие другого экземпляра среды разработки. При необходимости назначить разные порты и подключить агент к адресу нужного экземпляра.
Флажок включён, но сервер отключён окружением	Проверить значение `AGAVA_DEVENV_MCP_PORT`; значение `0` запрещает запуск.

В состоянии указано неверное значение порта	Задать допустимое число и перезапустить среду разработки с исправленным окружением.
Выдано предупреждение о рабочей папке	Сопоставить путь открытого проекта с заданием. Выбрать нужный проект до начала изменений.
Команды отклоняются из-за модального окна	Ответить на диалог DevEnv или закрыть его, затем повторить команду. Это относится и к открытому окну настроек.
Среда разработки выполняет предыдущий запрос	Дождаться завершения операции. Если открыт диалог, принять требуемое решение в среде разработки.
При удалении узла ожидается решение пользователя	Рассмотреть диалог о существующих ссылках и выбрать действие. Команда завершается после ответа.
Отклонена ссылка на переменную или функцию скрипта	Использовать допустимый узел или сигнал либо включить соответствующее разрешение в настройках, если такая связь предусмотрена проектом.
Не удаётся создать AGENTS.md	Проверить, что открыт проект `.asp`, известен его путь и папка доступна для записи.
Изменения отсутствуют после повторного открытия проекта	Проверить выполнение сохранения проекта. Отдельно учитывать сразу записываемые файлы документов.
RTU использует прежний вариант программы или окна	Повторно выполнить `runtime_start` после изменения проекта.
Команды исследования RTU недоступны	Проверить успешность запуска через `runtime_start`, состояние подключения и сообщения о загрузке.
Проверка прошла, но поведение отличается от ожидаемого	Проверить сценарий выполнения, входные значения, привязки, период задач и сообщения среды исполнения.

Модальные диалоги среды разработки требуют ответа пользователя. Инструмент `runtime_dialog` относится к диалогам запущенного RTU и не закрывает окно настроек или другие модальные окна среды разработки.