

Функциональное задание: Система скоринга токенов Solana (To The Moon)

Версия: 10.0

Дата: 09.09.2025

1. Введение и цели

Данный документ описывает функциональные требования к веб-приложению "**To The Moon**", предназначенному для автоматического отслеживания, анализа и скоринга криптовалютных токенов, которые успешно прошли миграцию с платформы Pump.fun на децентрализованные биржи (DEX) в блокчейне Solana.

Основная цель системы — предоставлять публичный, актуальный, отсортированный список перспективных токенов, основываясь на гибко настраиваемой системе скоринга.

Ключевое условие: система отслеживает **только те токены**, которые имеют пул ликвидности **WSOL/Токен**.

2. Архитектура системы

Система будет состоять из трех основных компонентов: бэкенд-приложение, база данных и фронтенд-интерфейс.

1. Бэкенд-приложение (Backend Application):

- Серверное приложение (например, на Python/FastAPI, Node.js/Express), отвечающее за всю бизнес-логику.
- WebSocket Worker:** Отдельный, постоянно работающий процесс, который устанавливает и поддерживает соединение с `wss://pumpportal.fun/api/data`.
- REST API:** Предоставляет набор публичных эндпоинтов для взаимодействия с фронтендом.

2. База данных (Database):

- PostgreSQL** в качестве основной реляционной СУБД.

3. Планировщик задач (Task Scheduler):

- Используется системный cron или интегрированная в фреймворк библиотека (например, Celery Beat) для выполнения периодических задач.

4. Веб-интерфейс (Frontend):

- Современное веб-приложение**, разработанное на React, Vue или Svelte. Приложение будет состоять из нескольких разделов, как минимум: главный экран (Дашборд) и страница настроек.
- Полностью публичный доступ.
- Размещается по адресу <https://tothemoon.sh1z01d.ru/>.

3. Алгоритм сбора данных и скоринга

Шаг 1: Отслеживание миграции и первичное сохранение

1. **WebSocket Worker** на бэкенде устанавливает и поддерживает WebSocket-соединение с `wss://pumpportal.fun/api/data`.
2. Воркер подписывается **только на одно событие**: `{"method": "subscribeMigration"}`.
3. При получении события `migration`, система извлекает из него `mint` адрес, **название** (`name`) и **символ** (`symbol`) токена.
4. Система **немедленно создает запись** о токене в таблице `tokens` с промежуточным статусом `monitoring`.

Шаг 2: Валидация и активация

1. **Периодическая** задача (запускаемая, например, каждую минуту) ищет в базе данных все токены со статусом `monitoring`.
2. Для каждого такого токена она выполняет GET-запрос к API DexScreener: `https://api.dexscreener.com/token-pairs/v1/solana/{mint}`.
3. Система анализирует ответ и проверяет выполнение **всех** следующих условий:
 - **Наличие целевого пула на Pump.fun**: В ответе должна быть хотя бы одна пара, где `baseToken.address` — это `mint` токена, `quoteToken.symbol` — это `"WSOL"`, и `dexId` — это `"pumpfun-amm"`.
 - **Наличие внешнего пула**: В ответе должна быть хотя бы одна пара с `dexId`, отличным от `"pumpfun-amm"`.
4. **Результаты проверки**:
 - **Если все условия выполнены**: статус токена в базе данных меняется с `monitoring` на `active`.
 - **Если условия не выполнены**: токен остается в статусе `monitoring` для проверки в следующем цикле.

Шаг 3: Извлечение признаков для скоринга

Для токенов со статусом `active` **периодическая задача обновления скоров** собирает следующие данные:

1. **Общая ликвидность (L_tot)**: Суммируется поле `liquidity.usd` **только для пулов, где** `quoteToken.symbol` равен `"WSOL"`.
2. **Изменение цены (ΔP_5m, ΔP_15m)**: Извлекаются значения из `priceChange` **только для пулов WSOL/токен**.
3. **Количество транзакций (N_5m)**: Суммируются `buys` и `sells` из `txns` **только для пулов WSOL/токен** за 5-минутный интервал.
4. **Количество держателей (HD)**: Выполняется запрос к API Helius.

Шаг 4: Расчет сора (Score)

Расчет производится на бэкенде по формулам:

1. **Нормирование метрик**:
 - $LIQ = \log_{10}(L_tot)$

- $I = \text{clip}((LIQ - 4) / 2)$
 - $SV = \text{abs}(\Delta P_{5m})$
 - $s = \text{clip}(SV / 0.1)$
 - $MV = \text{abs}(\Delta P_{5m}) / (\text{abs}(\Delta P_{15m}) + 0.001)$
 - $m = \text{clip}(MV)$
 - $TF = N_{5m} / 5$
 - $t = \text{clip}(TF / 300)$
 - $HD_{\text{norm}} = \text{clip}(\log_{10}(HD) / 4)$
2. Итоговая формула:
- $$\text{Score} = HD_{\text{norm}} * (W_s * s + W_I * I + W_m * m + W_t * t)$$

Шаг 5: Жизненный цикл и архивация

1. **Частота обновлений:** Настраивается в веб-интерфейсе для "горячих" и "остывших" active токенов.
2. **Архивация токенов: Задача обслуживания** (запускаемая cron, например, каждый час) выполняет две проверки:
 - **Архивация по низкому скору:** Ищет токены со статусом active, чей score держится ниже `min_score` в течение периода, заданного в настройках (по умолчанию 12 часов), и меняет их статус на archived.
 - **Архивация по тайм-ауту мониторинга:** Ищет токены со статусом monitoring, которые были созданы (`created_at`) более 12 часов назад, и меняет их статус на archived.

5. Схема базы данных (PostgreSQL)

Таблица 1: tokens

Поле	Тип	Описание
id	BIGSERIAL, PK	Уникальный идентификатор записи
mint_address	TEXT, UNIQUE	Адрес контракта токена
name	TEXT	Название токена
symbol	TEXT	Символ (тикер) токена
status	VARCHAR(20)	Текущий статус ('monitoring', 'active', 'archived')
created_at	TIMESTAMPTZ	Время создания записи (миграции)
last_updated_at	TIMESTAMPTZ	Время последнего обновления данных

Таблица 2: token_scores

Поле	Тип	Описание
id	BIGSERIAL, PK	Уникальный идентификатор записи
token_id	BIGINT, FK	Ссылка на tokens.id
score	NUMERIC(10, 4)	Итоговый рассчитанный скор
metrics	JSONB	Объект со всеми метриками (L_tot, ΔP_5m и т.д.)

| created_at | TIMESTAMPTZ | Время расчета данного сгора |

Таблица 3: app_settings

| Поле | Тип | Описание |

| :----- | :----- | :----- |

| key | TEXT, PK | Ключ настройки (e.g., 'weight_s', 'min_score') |

| value | TEXT | Значение настройки |

6. Требования к веб-интерфейсу

6.0. Общие требования к интерфейсу

Веб-интерфейс должен быть эстетически привлекательным, интуитивно понятным и удобным в использовании (юзабельным). Все элементы управления, поля ввода и отображаемые данные должны сопровождаться пояснительными текстами, всплывающими подсказками (тултипами) или описаниями, чтобы пользователь без труда понимал назначение каждой функции.

6.1. Главный экран (Дашборд)

- Отображает таблицу с токенами, у которых статус active и скор выше min_score.
- **Сортировка:** По умолчанию по скору (убывание).
- **Пагинация:** Применяется, если результатов больше 50 на странице.
- **Структура таблицы:**

Название (Символ)	Скор	Ликвиднос ть (USD)	Δ Цены (5м / 15м)	Транзакции (5м)	Холдеры	Solscan
Myro (\$MYRO)	0.8750	\$1,234,567	+5.2% / +12.8%	450	12,345	Просмотр

- **Пояснения к ссылкам:**
 - **Solscan:** Прямая ссылка на страницу токена на Solscan, формируемая по шаблону: `https://solscan.io/token/{mint_address}`.

6.2. Страница настроек

Раздел для полного контроля над поведением системы. Каждое поле должно иметь понятное название и поясняющую подпись.

Форма "Весовые коэффициенты"

- **Вес волатильности (W_s):** [0.35]
 - *Подсказка:* Определяет, насколько важна краткосрочная активность цены токена. Более высокое значение отдает приоритет токенам с резкими изменениями цены.
- **Вес ликвидности (W_l):** [0.25]
 - *Подсказка:* Влияние общего объема ликвидности в пулах WSOL. Высокое значение помогает отбирать более "стабильные" и крупные токены.
- **Вес импульса (W_m):** [0.20]

- Подсказка: Оценивает ускорение цены (сравнение динамики за 5 и 15 минут). Высокий вес у этого параметра помогает находить токены в начале фазы активного роста.
- **Вес частоты торгов (W_t): [0.20]**
 - Подсказка: Учитывает количество сделок за интервал. Показывает "интерес" и вовлеченность трейдеров. Высокое значение отдает приоритет токенам с активной торговлей.

Форма "Пороги"

- **Минимальное значение скоры (τ): [0.1]**
 - Подсказка: Токены, чей скор ниже этого значения, не будут отображаться на главной странице.

Форма "Тайминги и жизненный цикл"

- **Интервал для "горячих" токенов (сек): [10]**
 - Подсказка: Как часто обновлять данные для токенов с высоким скором. Слишком низкое значение может привести к превышению лимитов API, слишком высокое — к потере актуальности данных. Рекомендуемое значение: 5-15.
- **Интервал для "остывших" токенов (сек): [45]**
 - Подсказка: Как часто обновлять данные для токенов со стабильно низким скором. Позволяет экономить ресурсы и не превышать лимиты API. Рекомендуемое значение: 30-60.
- **Период неактивности для архивации (час): [12]**
 - Подсказка: Если скор активного токена держится ниже минимального порога дольше указанного времени, токен будет заархивирован и перестанет отслеживаться. Это помогает очищать список от "мертвых" монет.

Кнопка "Сохранить и Пересчитать"

- Сохраняет все настройки в базу данных через API бэкенда.
- Отправляет команду