

# Техническое задание: Встраивание приложений в Element X Web

---

**Версия:** 1.0 **Дата:** 30 января 2026 **Основано на:** iOS реализации Element X Fork

---

## 1. Обзор

---

### 1.1 Цель

Добавить функциональность встраиваемых приложений (виджетов) в веб-версию Element X, аналогичную реализованной в iOS-версии. Виджеты являются независимыми веб-приложениями, отображаемыми внутри клиента Matrix.

### 1.2 Ключевые возможности

- Каталог доступных приложений
  - Встраивание веб-приложений через iframe
  - Категоризация и поиск приложений
  - Запись звонков с возможностью прослушивания
  - Интеграция с Matrix аутентификацией
- 

## 2. Архитектура

---

### 2.1 Структура навигации

Главный экран (TabBar)

- └── Контакты
- └── Звонки (с историей и записями)
- └── Чаты
- └── Приложения ← НОВЫЙ РАЗДЕЛ
  - └── Список приложений
    - |     └── Поиск
    - |     └── Фильтры по категориям
    - |     └── Карточки приложений
  - └── Экран приложения (WebView/iframe)

### 2.2 Компоненты

```

src/
  └── components/
    ├── AppsTab/
    |   ├── AppsList.tsx          # Список приложений
    |   ├── AppCard.tsx           # Карточка приложения
    |   ├── AppCategories.tsx    # Фильтр категорий
    |   ├── AppWebView.tsx        # Контейнер для iframe
    |   └── AppSearchBar.tsx      # Поиск
    └── CallsTab/
        ├── CallHistory.tsx       # История звонков
        ├── CallHistoryItem.tsx    # Элемент истории
        └── RecordingPlayer.tsx    # Плеер записей
    └── common/
        └── LoadingIndicator.tsx  # Индикатор загрузки
  └── services/
    ├── AppsService.ts           # Сервис работы с приложением
    └── RecordingService.ts      # Сервис записи звонков
  └── models/
    ├── AppItem.ts               # Модель приложения
    ├── CallHistoryItem.ts       # Модель записи звонка
    └── RecordingState.ts        # Состояние записи
  └── stores/
    ├── appsStore.ts             # Стор приложений
    └── callsStore.ts            # Стор звонков

```

---

### 3. Модели данных

#### 3.1 AppItem (Приложение)

```

interface AppItem {
  id: string;
  name: string;
  description: string;
  icon: string;           // SF Symbols / Material Icons
  url: string;            // URL приложения
  category: AppCategory;
  requiredPermissions?: string[];
}

enum AppCategory {
  ALL = "all",
  PRODUCTIVITY = "productivity",
  COMMUNICATION = "communication",
  TOOLS = "tools",
  ANALYTICS = "analytics"
}

```

### 3.2 CallHistoryItem (История звонков)

```
interface CallHistoryItem {
    id: string;
    contactName: string;
    contactId: string;      // Matrix User ID
    callType: CallType;
    timestamp: Date;
    duration?: number;       // в секундах
    isMissed: boolean;
    recordingURL?: string;  // URL записи (если есть)
}

enum CallType {
    INCOMING = "incoming",
    OUTGOING = "outgoing",
    VIDEO = "video"
}
```

### 3.3 RecordingState (Состояние записи)

```
type RecordingState =
| { type: "idle" }
| { type: "starting" }
| { type: "recording"; egressId: string }
| { type: "stopping" }
| { type: "error"; message: string };
```

---

## 4. API Endpoints

---

### 4.1 Apps API

Базовый URL: <https://api.{domain}/api/apps>

**GET /api/apps/list**

Получение списка доступных приложений.

**Response:**

```
{  
    "success": true,  
    "apps": [  
        {  
            "id": "stats-app",  
            "name": "Статистика",  
            "description": "Аналитика и отчеты",  
            "icon": "chart.bar.fill",  
            "url": "https://stats.market.implica.ru",  
            "category": "analytics"  
        }  
    ]  
}
```

## 4.2 Recording API

**Базовый URL:** <https://api.{domain}/api/recording>

### POST /api/recording/start

Начать запись звонка.

#### Request:

```
{  
    "roomName": "!roomId:server.com"  
}
```

#### Response:

```
{  
    "success": true,  
    "egressId": "EG_xxxxx",  
    "status": 1,  
    "roomName": "!roomId:server.com",  
    "filepath": "recordings/room_timestamp.mp4"  
}
```

### POST /api/recording/stop

Остановить запись.

#### Request:

```
{  
    "egressId": "EG_xxxxx"  
}
```

**Response:**

```
{  
    "success": true,  
    "egressId": "EG_xxxxx",  
    "status": 2  
}
```

**GET /api/recording/status/:egressId**

Получить статус записи.

**Response:**

```
{  
    "success": true,  
    "egress": {  
        "egressId": "EG_xxxxx",  
        "roomName": "!roomId:server.com",  
        "status": 1,  
        "startedAt": "2026-01-30T10:00:00Z",  
        "endedAt": null  
    }  
}
```

**GET /api/recording/list**

Список всех записей.

**Query params:** ?roomName=!roomId:server.com (опционально)

**Response:**

```
{  
    "success": true,  
    "recordings": [  
        {  
            "egressId": "EG_xxxxx",  
            "roomName": "!roomId:server.com",  
            "status": 2,  
            "startedAt": "2026-01-30T10:00:00Z",  
            "endedAt": "2026-01-30T10:30:00Z",  
            "downloadUrl": "https://storage.../recording.mp4"  
        }  
    ]  
}
```

---

## 5. UI Компоненты

### 5.1 Список приложений (AppsList)

#### Требования:

- Заголовок "Приложения" с большим стилем (collapsible)
- Поле поиска под заголовком
- Горизонтальный скролл с чипами категорий
- Список карточек приложений
- Skeleton-загрузка при ожидании данных
- Empty state при отсутствии приложений

#### Макет карточки:

[Icon]	Название приложения	→	
52x52	Описание приложения		
	(до 2 строк)		

#### Категории (FilterChips):

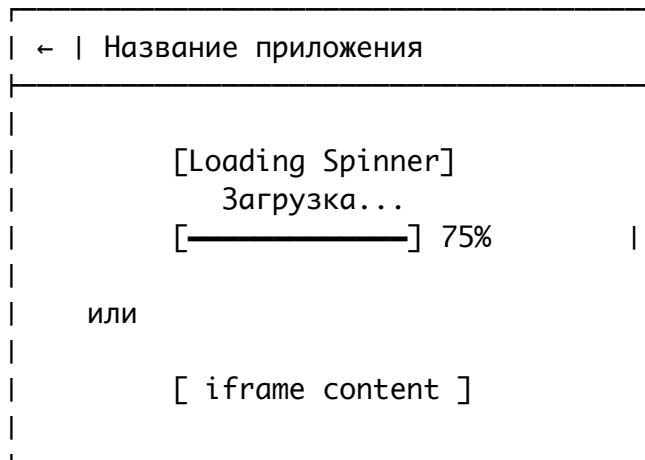
[ Все ] [ Продуктивность ] [ Связь ] [ Инструменты ]

### 5.2 WebView приложения (AppWebView)

#### Требования:

- Заголовок с названием приложения (inline)
- Кнопка "Назад"
- Индикатор загрузки с прогрессом
- iframe с полноэкранным режимом
- Обработка ошибок загрузки

**Структура:**

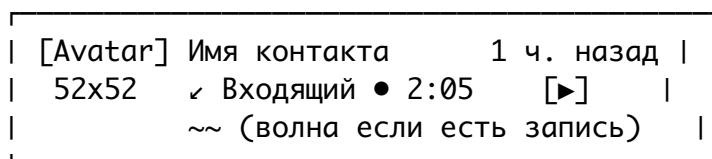


### 5.3 История звонков (CallHistory)

**Требования:**

- Заголовок "Звонки"
- Фильтры: Все | Пропущенные | Входящие | Исходящие
- Список звонков с аватарами
- Кнопка воспроизведения записи (если есть)
- Относительное время (1 ч. назад, вчера)

**Макет элемента:**



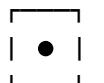
### 5.4 Кнопка записи на экране звонка

**Состояния:**

- Idle: серый круг с иконкой записи
- Starting/Stopping: spinner

- Recording: красный круг с иконкой stop + индикатор "REC"
- Error: оранжевый круг

Макет:



← Idle (серый)



← Recording (красный) + [REC]

---

## 6. Сервисы

---

### 6.1 AppsService

```
class AppsService {  
    private baseURL: string;  
  
    async getAppsList(): Promise<AppItem[]>;  
    async getAppById(id: string): Promise<AppItem | null>;  
    async getAppsByCategory(category: AppCategory): Promise<AppI  
}
```

### 6.2 RecordingService

```
class RecordingService {  
    private baseURL: string;  
    private stateSubject: BehaviorSubject<RecordingState>;  
  
    get state$(): Observable<RecordingState>;  
    get currentState(): RecordingState;  
  
    async startRecording(roomName: string): Promise<string>; //  
    async stopRecording(): Promise<void>;  
    async getStatus(egressId: string): Promise<EgressInfo>;  
    async listRecordings(roomName?: string): Promise<RecordingIr  
}
```

---

## 7. Интеграция

---

## 7.1 Аутентификация для iframe

При загрузке приложения в iframe передавать Matrix access token через:

### Вариант 1: URL параметр

```
https://app.example.com?access_token=xxx&user_id=@user:server.
```

### Вариант 2: PostMessage API

```
iframe.contentWindow.postMessage({
  type: 'matrix_auth',
  accessToken: 'xxx',
  userId: '@user:server.com',
  homeserverUrl: 'https://matrix.server.com'
}, '*');
```

## 7.2 Конфигурация

Добавить в настройки приложения:

```
interface AppConfig {
  appsApiBaseUrl: string;      // https://api.market.implica.
  recordingApiBaseUrl: string; // https://api.market.implica.
  enableAppsTab: boolean;     // true
  enableCallRecording: boolean; // true
}
```

## 7.3 Well-known Matrix конфигурация

Добавить в .well-known/matrix/client:

```
{
  "m.homeserver": {
    "base_url": "https://matrix.server.com"
  },
  "io.element.apps": {
    "api_url": "https://api.server.com/api/apps"
  },
  "io.element.recording": {
    "api_url": "https://api.server.com/api/recording"
  }
}
```

## 8. Безопасность

---

### 8.1 Content Security Policy для iframe

Content-Security-Policy:

```
frame-src 'self' https://*.implica.ru https://stats.market.i  
frame-ancestors 'self';
```

### 8.2 Sandbox атрибуты iframe

```
<iframe  
    src="..."  
    sandbox="allow-scripts allow-same-origin allow-forms allow-p  
    allow="camera; microphone; fullscreen"  
/>
```

### 8.3 Валидация URL приложений

- Проверять URL на whitelist доменов
  - Не разрешать data: и javascript: схемы
  - Проверять HTTPS для production
- 

## 9. Локализация

---

### 9.1 Русский (по умолчанию)

```
{  
    "apps.title": "Приложения",  
    "apps.search.placeholder": "Поиск приложений",  
    "apps.category.all": "Все",  
    "apps.category.productivity": "Продуктивность",  
    "apps.category.communication": "Связь",  
    "apps.category.tools": "Инструменты",  
    "apps.empty.title": "Нет приложений",  
    "apps.empty.description": "Приложения будут отображаться здесь",  
    "apps.loading": "Загрузка...",  
  
    "calls.title": "Звонки",  
    "calls.filter.all": "Все",  
    "calls.filter.missed": "Пропущенные",  
    "calls.filter.incoming": "Входящие",  
    "calls.filter.outgoing": "Исходящие",  
    "calls.type.incoming": "Входящий",  
    "calls.type.outgoing": "Исходящий",  
    "calls.type.video": "Видеозвонок",  
    "calls.type.missed": "Пропущенный",  
  
    "recording.start": "Начать запись",  
    "recording.stop": "Остановить запись",  
    "recording.consent.title": "Начать запись?",  
    "recording.consent.message": "Этот звонок будет записан. Всё?",  
    "recording.consent.confirm": "Начать запись",  
    "recording.consent.cancel": "Отмена"  
}
```

---

## 10. Тестирование

### 10.1 Unit тесты

- AppsService: загрузка списка, фильтрация, обработка ошибок
- RecordingService: старт/стоп записи, обработка состояний
- Компоненты: рендеринг, взаимодействие

### 10.2 Integration тесты

- Загрузка приложения в iframe
- Передача аутентификации
- Запись и воспроизведение звонка

### 10.3 E2E тесты

- Навигация между табами

- Поиск и фильтрация приложений
  - Полный цикл записи звонка
- 

## 11. Примеры реализации (Reference)

---

### iOS реализация

Компонент	Файл
Список приложений	WidgetsListScreen.swift
Модели	WidgetsListScreenModels.swift
WebView	WidgetWebViewScreen.swift
История звонков	CallsListScreen.swift
Сервис записи	RecordingService.swift
Модели записи	RecordingModels.swift
Кнопка записи	RecordingButton.swift

### Репозиторий

<https://github.com/example/element-x-fork>  
└── ios/ElementX/Sources/Screens/WidgetsListScreen/  
└── ios/ElementX/Sources/Screens/WidgetWebViewScreen/  
└── ios/ElementX/Sources/Screens/CallsListScreen/  
└── ios/ElementX/Sources/Services/Recording/  
└── backend/recording-api/

---

## 12. План реализации

---

### Фаза 1: Базовая структура (1 неделя)

- Создать компоненты AppsTab
- Добавить навигацию в TabBar
- Реализовать AppsService
- Создать базовый AppWebView

### Фаза 2: Звонки и записи (1 неделя)

- Создать компоненты CallsTab
- Реализовать RecordingService

- Добавить кнопку записи на экран звонка
- Интегрировать плеер записей

### Фаза 3: Интеграция и тестирование (1 неделя)

- Настроить CSP и sandbox
- Реализовать передачу аутентификации
- Написать тесты
- Провести code review

### Фаза 4: Деплой и документация

- Обновить конфигурацию сервера
- Развернуть Recording API
- Написать пользовательскую документацию

---

## 13. Ссылки

---

- [Matrix Widget API](#)
- [LiveKit Egress](#)
- [Element Call](#)
- [Compound Design System](#)