|  |
| --- |
| Техническое приложение к регламенту информационного взаимодействия |
| ОПИСАНИЕ ИНТЕГРАЦИОННЫХ ПРОФИЛЕЙПОДСИСТЕМА УПРАВЛЕНИЯ ПОТОКАМИМОДУЛЬ ПРОФИЛЕЙ ОБСЛУЖИВАНИЯ |

История версий

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Версия | Дата | Автор | Примечание |
| 1 | 1.0 | 22/04/2023 | Туркин А. А. | Первая версия регламента |

Содержание

[1 Подключение 4](#_Toc101795703)

[2 Объявление доступных профилей обслуживания 5](#_Toc101795704)

[2.1 Работа с расписанием 5](#_Toc101795705)

[2.2 Формирование заявки 5](#_Toc101795706)

[2.3 Выполнение заявки 6](#_Toc101795707)

[2.4 Ролевой контекст 6](#_Toc101795708)

[3 Добавление виджета расписания как библиотеки 7](#_Toc101795709)

[3.1 Добавление виджета расписания как iframe 8](#_Toc101795710)

[3.2 Добавление виджета заявок как JS-библиотеку 9](#_Toc101795711)

[3.3 Добавление виджета заявок как iframe 12](#_Toc101795712)

Аннотация

Настоящий документ описывает регламент работы с сервисом интеграционной платформы N3 «Подсистемы Управление потоками» (Подсистемы Управление потоками) в части формирования и использования виджетов для работы с пользовательским интерфейсом подсистемы Телемедицинских консультаций.

Указанный сервис обеспечивает механизмы взаимодействия и обмена данными между различными информационными системами, обслуживающими процесс передачи сведений о профилях обслуживания в рамках конкретной бизнес-логики.

В состав описания включены порядок подключения к сервису, формирования настроек предварительных работ со стороны медицинской организации, инструкции по интеграции виджетов работы с заявками и доступными профилями обслуживания.

# Подключение

Для подключения можно работать с системой через API (<http://api.n3med.ru/api/szpv/reglament-po-telemeditsine-spb/>) или интегрировать виджеты работы с заявками и расписание. Инструкции по подключению виджетов приведены ниже.

Для запуска виджета работы с заявками понадобится сформировать и передать для пользователя ролевой контекст. Для запуска виджета работы с доступными профилями и расписанием в него передаётся токен доступа к подсистеме tm.schedule.

Токены доступа к тестовому и опытному стенду к подсистеме ТМК выдаются по запросу в техническую поддержку разработчика.

Так же при обращении за подключением к системе медицинская организация должна определиться для себя, где она будет вести расписание врачей по работе с телемедицинскими заявками. Допускается вести расписание в МИС или в модуле tm.schedule через виджет. Расписание сформированное через виджет будут так же передаваться с помощью сервиса записи (http://api.n3med.ru/api/szpv/zapis-na-priem-k-vrachu-po-fhir-arkhitekture/).

# Объявление доступных профилей обслуживания

Для предоставления доступа к подаче заявки в свою организацию в системе предусмотрена работа с доступными профилями обслуживания. Запустив виджет расписания с токеном организации, пользователь для неё может создавать и редактировать описание профилей обслуживания, которые имеют набор характеристик, описывающих первичные условия приёма заявки.

Характеристики соответственно играют в бизнес логике системы следующие роли:

1. Возрастные категории – указывают на специфику оказания медицинской консультации
2. Номенклатура медицинских услуг – необходимо указать для работы с сервисом записи
3. Составной профиль медицинской помощи – указывает на ключевую характеристику, с которой пользователь начнёт подбор медицинской организации для подачи заявки
4. Тип предоставляемой услуги – поможет отсечь пациентов, которые не должны воспользоваться профилем организации, если не имеют в ней прикрепления по полису ОМС
5. Маршрут ТМК – бизнес-процесс в подсистеме Управление потоками, по которому предоставляется телемедицинская консультация

## Работа с расписанием

Медицинская организация способна вести расписание в своей МИС или в виджете доступных профилей и расписания.

Для формирования расписания в виджете пользователю потребуется выполнить следующий порядок действий:

1. Создать доступный профиль обслуживания
2. Создать шаблон сетки расписания на неделю в формате очереди или талонов. Очередь определяет объём принимаемых заявок на неделю, врач может исполнять их в любое время. Талоны указывают на конкретный промежуток времени в одном дне на который чаще всего настраивается возможность записать одного пациента.
3. Добавить врача, который будет работать с ТМК. Врача в систему возможно добавить только по СНИЛС из справочника ФРМР НСИ.
4. Добавить расписание для профиля, врача и шаблона. При сохранении расписания согласно шаблону для указанного врача будут созданы доступные слоты, которые будет возможно посмотреть в разделе виджета «Календарь».

Созданное таким образом расписание при условии, что оно объявлено в СЗПВ как источник расписания для ТМК будет передаваться пациенту, или направляющему врачу, или диспетчеру целевой в ходе работы над заявкой.

## Формирование заявки

Для формирования заявки потребуется подключить виджет работы с заявками для конкретного пользователя (врач, диспетчер регистратуры). Для подбора данных в виджете работы с заявкой используется ролевой контекст, передаваемый в виджет при его инициализации. Примеры ролевых контекстов указаны ниже.

1. Для создания заявки потребуется выполнить следующий порядок действий:
2. Открыть виджет на вкладке «Создать заявку»
3. Выбрать интересующий составной профиль обслуживания
4. Далее выбрать организацию, а на самом деле профиль созданный предварительно целевой организацией

Заполнить форму и нажать кнопку «Сохранить»

## Выполнение заявки

Для работы с заявкой со стороны целевой организации так же потребуется передать в виджет работы с заявкой ролевой контекст пользователя. В зависимости от ролевого контекста виджет отобразить соответствующие данные.

Пользователям так же будут доступны вкладки «Заявки», «Создать заявку», «Календарь».

Обрабатывать заявки пользователи смогут согласно настройкам бизнес-логике настроенной для неё со стороны системы ТМК.

## Ролевой контекст

|  |  |  |  |
| --- | --- | --- | --- |
| Роль | Идентификатор роли | Параметры | Пример |
| Врач | 5ff16ba7-9edd-41a4-bd06-2ee33cfd4597 | organization – идентификатор организации по справочнику 1.2.643.2.69.1.1.1.64SNILS – СНИЛС врача | { "roleContext": { "e34d94b3-e889-49fe-96a0-65194af7cb94": { "organization": "fb3f350b-22c7-4cc6-9dc1-e257b3175d25" } }} |
| Диспетчер | e34d94b3-e889-49fe-96a0-65194af7cb94 | organization – идентификатор организации по справочнику 1.2.643.2.69.1.1.1.64 | { "roleContext": { "5ff16ba7-9edd-41a4-bd06-2ee33cfd4597": { "organization": "fb3f350b-22c7-4cc6-9dc1-e257b3175d25", "SNILS":"00011122233" } }} |
| Главный врач | 59576ab6-8c98-49df-81e7-e7789caf02db | organization – идентификатор организации по справочнику 1.2.643.2.69.1.1.1.64 | { "roleContext": { "59576ab6-8c98-49df-81e7-e7789caf02db ": { "organization": "fb3f350b-22c7-4cc6-9dc1-e257b3175d25", "SNILS":"00011122233" } }} |

# Добавление виджета расписания как библиотеки



React
Для интеграции виджета необходимо добавить в проект компонент MicroFrontend
Параметры передаваемые в компонент:
name - имя компонента(TMSchedule)
host - адреc где развернут виджет (<base\_url>tm.schedule/)
history - компонент дающий доступ к истории браузера из библиотека "react-router-dom"

|  |
| --- |
| import { useEffect } from "react";const renderMicroFrontend = (name: string, history: any) => { // @ts-ignore window[`render${name}`](`${name}-container`, history);};export function MicroFrontend({ name, host, history }: any) { useEffect(() => { const scriptId = `micro-frontend-script-${name}`; if (document.getElementById(scriptId)) { renderMicroFrontend(name, history); return; } fetch(`${host}/asset-manifest.json`) .then((res) => res.json()) .then((manifest) => { const script = document.createElement("script"); script.id = scriptId; script.crossOrigin = ""; const hostElements = host.split('/'); const srcElements: string[] = []; manifest.files["main.js"]?.split('/')?.forEach((f: string) => { if (!hostElements.includes(f)) { srcElements.push(f); } }); script.src = `${host}/${srcElements.join('/')}`; script.onload = () => { renderMicroFrontend(name, history); }; script.id = 'micro-frontend-script'; document.head.appendChild(script);  if (manifest.files["main.css"]) { const link = document.createElement("link"); link.rel = "stylesheet"; const hrefElements: string[] = []; manifest.files["main.css"]?.split('/')?.forEach((f: string) => { if (!hostElements.includes(f)) { hrefElements.push(f); } }); link.href = `${host}/${hrefElements.join('/')}`; link.id = 'micro-frontend-style'; document.head.appendChild(link); } }); return () => { // @ts-ignore window[`unmount${name}`] && window[`unmount${name}`](`${name}-container`); document.getElementById('micro-frontend-style')?.remove(); document.getElementById('micro-frontend-script')?.remove(); }; }, []); return <main id={`${name}-container`} style={{width: '100%'}}/>;}  |

javascript

Добавить разметку для виджета на html странице

|  |
| --- |
| <main id="TMSchedule-container" style="width: 100%"/>  |

Добавить js код

|  |
| --- |
| <script type="text/javascript"> $(function () { setTimeout(() => { // Передача контекста в виджет window.postMessage({ isActualData: true, data: { token: "689ae90d-8779-4005-a443-1cee7d24e719", //token для запросов на бэк tm.schedule path: window.location.pathname // путь для роутинга } }, '\*'); }, 700); const host = 'http://r78-dev.zdrav.netrika.ru/tm.schedule/'; //адреc где развернут виджет const name = 'TMSchedule'; const scriptId = `micro-frontend-script-${name}`; fetch(`${host}/asset-manifest.json`) .then((res) => res.json()) .then((manifest) => { console.log(manifest); const script = document.createElement("script"); script.id = scriptId; script.crossOrigin = ""; const hostElements = host.split('/'); const srcElements = []; manifest.files["main.js"]?.split('/')?.forEach((f) => { if (!hostElements.includes(f)) { srcElements.push(f); } }); script.src = `${host}/${srcElements.join('/')}`; script.onload = () => { window[`render${name}`](`${name}-container`); }; script.id = 'micro-frontend-script'; document.head.appendChild(script);  if (manifest.files["main.css"]) { const link = document.createElement("link"); link.rel = "stylesheet"; const hrefElements = []; manifest.files["main.css"]?.split('/')?.forEach((f) => { if (!hostElements.includes(f)) { hrefElements.push(f); } }); link.href = `${host}/${hrefElements.join('/')}`; link.id = 'micro-frontend-style'; document.head.appendChild(link); } }); });</script>  |

## Добавление виджета расписания как iframe



Для добавления виджета нужно добавить код на html странице

|  |
| --- |
| <iframe title="tm-calendar" src="http://r78-dev.zdrav.netrika.ru/tm.schedule/" frameBorder="no" width="100%" height="100%" id="tm-calendar" /><script type="text/javascript"> $(function () { setTimeout(() => { // Передача контекста в виджет window.postMessage({ isActualData: true, data: { token: "689ae90d-8779-4005-a443-1cee7d24e719", //token для запросов на бэк tm.schedule path: window.location.pathname // путь для роутинга } }, '\*'); }, 700); });</script>  |

## Добавление виджета заявок как JS-библиотеку

react
Для интеграции виджета необходимо добавить в проект компонент MicroFrontend
Параметры передаваемые в компонент:
name - имя компонента (TMWidget)
host - адрес где развернут виджет (<base\_url>/tm.widgets/)
history - компонент дающий доступ к истории браузера из библиотека "react-router-dom"

|  |
| --- |
| import { useEffect } from "react";const renderMicroFrontend = (name: string, history: any) => { // @ts-ignore window[`render${name}`](`${name}-container`, history);};export function MicroFrontend({ name, host, history }: any) { useEffect(() => { const scriptId = `micro-frontend-script-${name}`; if (document.getElementById(scriptId)) { renderMicroFrontend(name, history); return; } fetch(`${host}/asset-manifest.json`) .then((res) => res.json()) .then((manifest) => { const script = document.createElement("script"); script.id = scriptId; script.crossOrigin = ""; const hostElements = host.split('/'); const srcElements: string[] = []; manifest.files["main.js"]?.split('/')?.forEach((f: string) => { if (!hostElements.includes(f)) { srcElements.push(f); } }); script.src = `${host}/${srcElements.join('/')}`; script.onload = () => { renderMicroFrontend(name, history); }; script.id = 'micro-frontend-script'; document.head.appendChild(script);  if (manifest.files["main.css"]) { const link = document.createElement("link"); link.rel = "stylesheet"; const hrefElements: string[] = []; manifest.files["main.css"]?.split('/')?.forEach((f: string) => { if (!hostElements.includes(f)) { hrefElements.push(f); } }); link.href = `${host}/${hrefElements.join('/')}`; link.id = 'micro-frontend-style'; document.head.appendChild(link); } }); return () => { // @ts-ignore window[`unmount${name}`] && window[`unmount${name}`](`${name}-container`); document.getElementById('micro-frontend-style')?.remove(); document.getElementById('micro-frontend-script')?.remove(); }; }, []); return <main id={`${name}-container`} style={{width: '100%'}}/>;}  |

javascript

Добавить разметку для виджета на html странице

|  |
| --- |
| <main id="TMWidget-container" style="width: 100%"/>  |

Добавить js код

|  |
| --- |
| <script type="text/javascript"> $(function () { setTimeout(() => { var roleContext = {}; roleContext["c7ec2fc0-982c-4ec1-ae84-c7ec1347ddfb"] = { idMPI: "689ae90d-8779-4005-a443-1cee7d24e719", idIP: "00000000-0000-0000-2222-000000000000" }; // Передача контекста роли пользователя в виджет window.postMessage({ isActualData: true, data: { roleContext: roleContext } }, '\*'); }, 700); const host = 'http://r78-dev.zdrav.netrika.ru/tm.widgets/'; //адреc где развернут виджет const name = 'TMWidget'; const scriptId = `micro-frontend-script-${name}`; fetch(`${host}/asset-manifest.json`) .then((res) => res.json()) .then((manifest) => { console.log(manifest); const script = document.createElement("script"); script.id = scriptId; script.crossOrigin = ""; const hostElements = host.split('/'); const srcElements = []; manifest.files["main.js"]?.split('/')?.forEach((f) => { if (!hostElements.includes(f)) { srcElements.push(f); } }); script.src = `${host}/${srcElements.join('/')}`; script.onload = () => { window[`render${name}`](`${name}-container`); }; script.id = 'micro-frontend-script'; document.head.appendChild(script);  if (manifest.files["main.css"]) { const link = document.createElement("link"); link.rel = "stylesheet"; const hrefElements = []; manifest.files["main.css"]?.split('/')?.forEach((f) => { if (!hostElements.includes(f)) { hrefElements.push(f); } }); link.href = `${host}/${hrefElements.join('/')}`; link.id = 'micro-frontend-style'; document.head.appendChild(link); } }); });</script>  |

На входе в виджет можно передавать данные:

|  |
| --- |
| "workflowId":"db648adb-aa34-4b08-a4b0-e4a11a6b9e44", //Идентификатор маршрута"transitionId":"c63356eb-092e-4b3d-a9e1-9059ae94bad6", //Идентификатор перехода создания заявки. Если его передать, то виджет запустится с экрана формы создания заявки"roleContext":{}, //ролевой контекст пользователя"formData":{} //передаются данные для заполнения предварительного заполнения формы  |

Пример контекста передаваемого в виджет:

|  |
| --- |
| {"workflowId":"db648adb-aa34-4b08-a4b0-e4a11a6b9e44","transitionId":"c63356eb-092e-4b3d-a9e1-9059ae94bad6","roleContext":{"c7ec2fc0-982c-4ec1-ae84-c7ec1347ddfb": { "idMPI": "689ae90d-8779-4005-a443-1cee7d24e719", "idIP":"00000000-0000-0000-2222-000000000000" }},"formData":{ "serviceRequest": { "reasonCode": "1", "сomplaints": "жалоба пациента", "urgency": "1", "performerOrganization": "00085ffb-1425-4420-bf43-025e4e815300", "ageCategory": "3", "REGIZ": { "iemk": "14,10181", "uo": "78204028479620,78204028479780", "odii": "ae526856-c1e3-46fe-8879-be1867d6677c,a474549e-0959-4c55-aef0-83f3bd3cedd8", "odli": "af4e6c21-67b9-4b07-8429-eea13d7fc7bf,a8cf4137-712f-40a1-a11b-13b4812fc2d9" }, "healthCareService": "02ef31bd-8173-495a-89ce-5694f13b8ae7", "typeOfServiceProvision": "1", "nomenclatureOfMedicalServices": "B01.029.001" }, "context": { "serviceRequest": { "category": "10" } }}} |

## Добавление виджета заявок как iframe

для автоматического перехода к карточке заявки введите в контексте
"processId":"<id процесса>"
для автоматического перехода к форме создания заявки введите в контексте
"transitionId ":"<id перехода>"
"workflowId ":"<id маршрута>"

Для добавления виджета нужно добавить код на html странице

|  |
| --- |
| <iframe title="tm-widget" src="http://r78-dev.zdrav.netrika.ru/tm.widgets/" // url виджета frameBorder="no" width="100%" height="100%" id="tm-widget" /><script type="text/javascript"> $(function () { setTimeout(() => { var roleContext = {}; roleContext["c7ec2fc0-982c-4ec1-ae84-c7ec1347ddfb"] = { idMPI: "689ae90d-8779-4005-a443-1cee7d24e719", idIP: "00000000-0000-0000-2222-000000000000" }; // Передача контекста роли пользователя в виджет window.postMessage({ isActualData: true, data: { roleContext: roleContext } }, '\*'); }, 700); });</script>  |