**утвержден**

643.МИФТ.00124-01 13 01-лу

**Система учета рабочего времени (Time Tracker)**

Описание программы

643.МИФТ.00124-01 13 01

Инв. № подл.

Подпись и дата

Взам. инв. №

Инв. № дубл.

Подпись и дата

Листов 15

2018

аннотация

Документ содержит описание ПО Системы учета рабочего времени (Time Tracker). Приводятся общие сведения о программе, ее функциональном назначении, входных, выходных данных.

содержание

[1 Общие сведения 4](#_Toc510575620)

[2 Функциональное назначение 5](#_Toc510575621)

[3 Описание логической структуры 6](#_Toc510575622)

[4 Используемые технические средства 8](#_Toc510575623)

[5 Вызов и загрузка 9](#_Toc510575624)

[6 Входные данные 10](#_Toc510575625)

[7 Выходные данные 11](#_Toc510575626)

8 Лист регистрации изменений.................................................................................... 15

# 1 Общие сведения

Программное обеспечение написано с использованием языка программирования Java, версия 8.

Сокращенное имя – ПО Time Tracker.

Внутреннее имя – TimeTracker.

Исполняемый файл – TimeTracker.jar.

Параметры исполняемого файла:

- Кросплатформенный исполняемый файл для виртуальной машины Java,

- Размер файла: ~40,5 Мб,

# 2 Функциональное назначение

ПО Time Tracker предназначено для учета рабочего времени сотрудников и реализует функции:

- Создание пользователей;

- удаление пользователей;

- регистрация событий прихода сотрудников на работу;

- регистрация событий ухода сотрудников с работы;

- формирование отчетов на заданный день, отображающих присутствие-отсутствие сотрудников, время прихода-ухода;

- формирование отчетов за указанный период для конкретных сотрудников.

Возможности ПО Time Tracker ограничены исключительно реализацией серверной части приложения и не предполагают никакого взаимодействия с конечными пользователями через графический интерфейс.

# 3 Описание логической структуры

Логическая структура ПО Time Tracker приведена на рисунке 1:

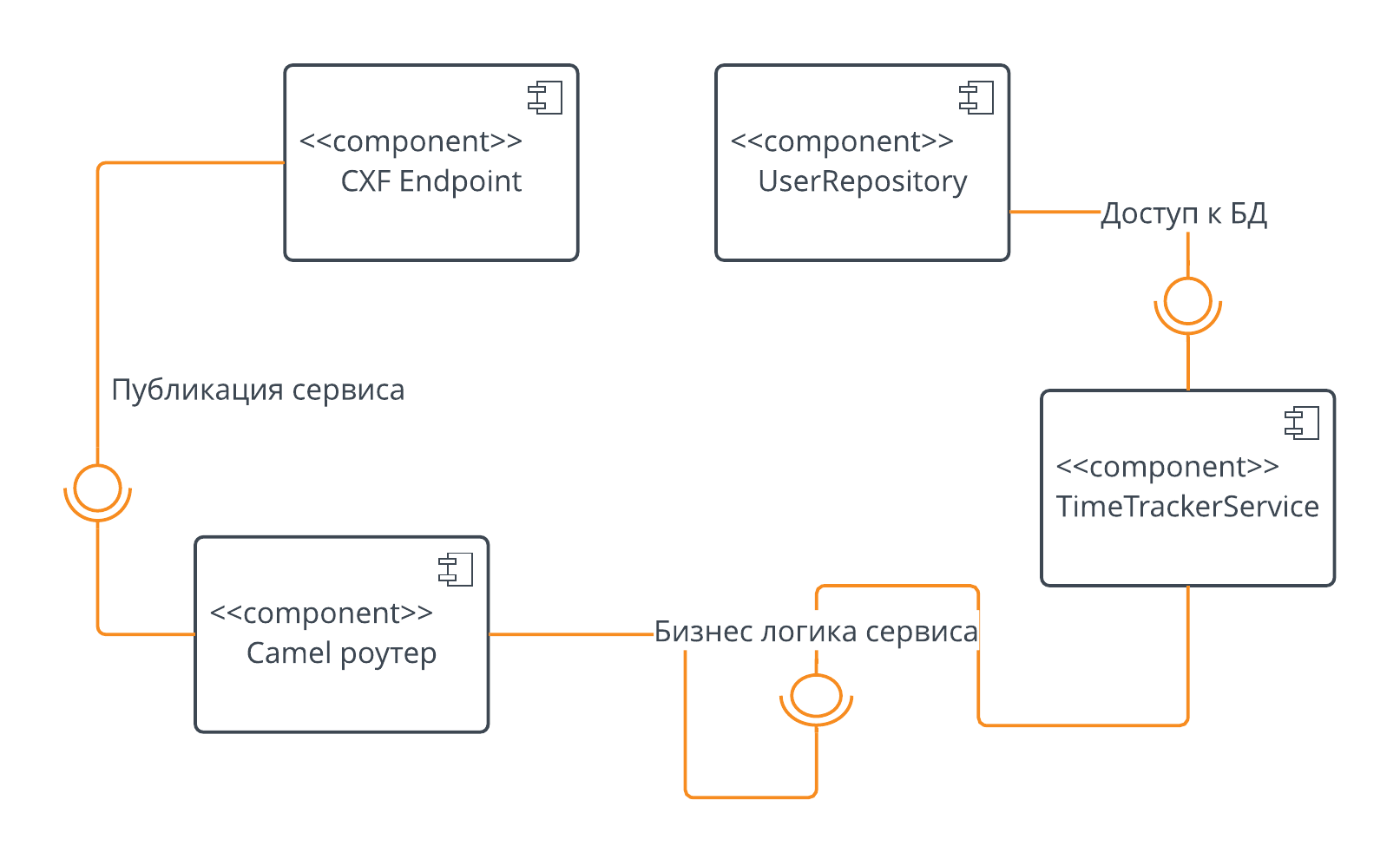


Рисунок 1 – Структурная схема ПО Time Tracker

К основным компонентам относятся:

1. UserRepository – слой доступа к сущностям приложения в БД, инкапсулирует в себе логику выборки данных.
2. CXF Endpoint – компонент из библиотеки Apache CXF, публикующий веб-сервис для доступа клиентских приложений и генерирующий схему сервиса (WSDL).
3. TimeTrackerService – бизнес-логика, реализующая основную функциональность сервиса.
4. Camel роутер – связующее звено приложения, выполняет маршрутизацию сообщений между конечными пользователями и внутренними компонентами.

Модель данных состоит из двух таблиц: пользователь-событие (user\_table – user\_event\_table). Связь между данными таблицами один ко многим из таблиц на основе первичного ключа id из таблицы user\_table и внешнего ключа user\_event\_table.



Рисунок 2 – Упрощенная модель БД

Принципиальная схема развертывания подразумевает наличие трех независимых серверов (виртуальных машин): веб-сервер с развернутым ПО Time Tracker, БД PostgreSQL и брокер JMS-сообщений (ActiveMQ).



Рисунок 3 – Диаграмма развертывания

# 4 Используемые технические средства

ПО Time Tracker является стандартным переносимым (кроссплатформенным) приложением для платформы Java и может работать под управлениями тех ЭВМ и устройств, которые являются совместимыми с платформой Java.

# 5 Вызов и загрузка

Загрузка приложения на жесткий диск используемой ЭВМ может осуществляться любым из доступных способов.

Возможно два сценария вызова приложения:

1. В вид самостоятельного приложения

Предварительно перед вызовом приложения необходимо провести стандартную установку виртуальной машины Java (версия 8), включая настройку переменных окружения для доступа к исполняемому файлу виртуальной машины из каталога вызова ПО Time Tracker.

Для работы приложения также следует предоставить права на открытие порта 8080, настройки для доступа к БД PostgreSQL и JMS-сервера ActiveMQ (подробная информация по подготовке и настройке к вызову содержится в Руководстве Администратора).

Вызов ПО осуществляется вручную (двойным нажатием на исполняемом файле), либо с помощью специально подготовленных скриптов запуска.

1. В виде веб-архива для разворачивания на сервере приложений.

Такая конфигурация потребует отдельной сборки приложения (см. рекомендации в Руководстве Администратора).

# 6 Входные данные

Входными данными для работы ПО Time Tracker являются сообщения, посылаемые через поставляемый веб-сервис со стандартным SOAP-интерфейсом. Формат сообщений должен соответствовать WSDL-схеме данных сервиса (см. Приложение №1).

# 7 Выходные данные

К выходными данным ПО Time Tracker можно отнести:

1. Ответные сообщения, посылаемые через поставляемый веб-сервис со стандартным SOAP-интерфейсом. Формат сообщений соответствуют WSDL-схеме данных сервиса (см. Приложение №1).
2. Диагностические сообщения о работе, событиях и ошибках ПО (лог приложения). Данные сообщения выводятся по умолчанию в консоль, либо могут быть перенаправлены для записи в файл при необходимости.

**Приложение №1. Схема данных веб-сервиса Time Tracker.**

<?xml version='1.0' encoding='UTF-8'?><wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://taskdata.com/timetracker/ws" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:ns1="http://schemas.xmlsoap.org/soap/http" name="TimeTrackerServiceService" targetNamespace="http://taskdata.com/timetracker/ws">

<wsdl:types>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://taskdata.com/timetracker/ws" attributeFormDefault="unqualified" elementFormDefault="unqualified" targetNamespace="http://taskdata.com/timetracker/ws">

<xs:element name="createDailyReport" type="tns:createDailyReport"/>

<xs:element name="deleteUser" type="tns:deleteUser"/>

<xs:element name="loginUser" type="tns:loginUser"/>

<xs:element name="logoutUser" type="tns:logoutUser"/>

<xs:complexType name="createDailyReport">

<xs:sequence>

<xs:element minOccurs="0" name="arg0" type="tns:createDailyReportRequest"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="createDailyReportRequest">

<xs:sequence>

<xs:element minOccurs="0" name="reportDate" type="xs:dateTime"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="createDailyReportResponse">

<xs:sequence>

<xs:element maxOccurs="unbounded" minOccurs="0" name="userEvents" nillable="true" type="tns:user"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="user">

<xs:sequence>

<xs:element minOccurs="0" name="id" type="xs:long"/>

<xs:element maxOccurs="unbounded" minOccurs="0" name="userEvents" nillable="true" type="tns:userEvent"/>

<xs:element minOccurs="0" name="username" type="xs:string"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="userEvent">

<xs:sequence>

<xs:element minOccurs="0" name="eventTimestamp" type="tns:localDateTime"/>

<xs:element minOccurs="0" name="eventType" type="tns:eventType"/>

<xs:element minOccurs="0" name="id" type="xs:long"/>

</xs:sequence>

</xs:complexType>

<xs:complexType final="extension restriction" name="localDateTime">

<xs:sequence/>

</xs:complexType>

<xs:complexType name="deleteUser">

<xs:sequence>

<xs:element minOccurs="0" name="arg0" type="tns:deleteUserRequest"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="deleteUserRequest">

<xs:sequence>

<xs:element minOccurs="0" name="username" type="xs:string"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="deleteUserResponse">

<xs:sequence>

<xs:element minOccurs="0" name="user" type="tns:user"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="logoutUser">

<xs:sequence>

<xs:element minOccurs="0" name="arg0" type="tns:logoutUserRequest"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="logoutUserRequest">

<xs:sequence>

<xs:element minOccurs="0" name="eventTimestamp" type="xs:dateTime"/>

<xs:element minOccurs="0" name="username" type="xs:string"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="logoutUserResponse">

<xs:sequence>

<xs:element minOccurs="0" name="user" type="tns:user"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="createUserRequest">

<xs:sequence>

<xs:element minOccurs="0" name="username" type="xs:string"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="createUserResponse">

<xs:sequence>

<xs:element minOccurs="0" name="user" type="tns:user"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="loginUser">

<xs:sequence>

<xs:element minOccurs="0" name="arg0" type="tns:loginUserRequest"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="loginUserRequest">

<xs:sequence>

<xs:element minOccurs="0" name="eventTimestamp" type="xs:dateTime"/>

<xs:element minOccurs="0" name="username" type="xs:string"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="loginUserResponse">

<xs:sequence>

<xs:element minOccurs="0" name="user" type="tns:user"/>

</xs:sequence>

</xs:complexType>

<xs:simpleType name="eventType">

<xs:restriction base="xs:string">

<xs:enumeration value="LOGIN"/>

<xs:enumeration value="LOGOUT"/>

</xs:restriction>

</xs:simpleType>

<xs:element name="createDailyReportResponse" nillable="true" type="tns:createDailyReportResponse"/>

<xs:element name="deleteUserResponse" nillable="true" type="tns:deleteUserResponse"/>

<xs:element name="logoutUserResponse" nillable="true" type="tns:logoutUserResponse"/>

<xs:element name="createUser" nillable="true" type="tns:createUserRequest"/>

<xs:element name="createUserResponse" nillable="true" type="tns:createUserResponse"/>

<xs:element name="loginUserResponse" nillable="true" type="tns:loginUserResponse"/>

</xs:schema>

</wsdl:types>

<wsdl:message name="createDailyReport">

<wsdl:part element="tns:createDailyReport" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:message name="createDailyReportResponse">

<wsdl:part element="tns:createDailyReportResponse" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:message name="deleteUserResponse">

<wsdl:part element="tns:deleteUserResponse" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:message name="deleteUser">

<wsdl:part element="tns:deleteUser" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:message name="logoutUserResponse">

<wsdl:part element="tns:logoutUserResponse" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:message name="createUser">

<wsdl:part element="tns:createUser" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:message name="createUserResponse">

<wsdl:part element="tns:createUserResponse" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:message name="loginUserResponse">

<wsdl:part element="tns:loginUserResponse" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:message name="logoutUser">

<wsdl:part element="tns:logoutUser" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:message name="loginUser">

<wsdl:part element="tns:loginUser" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:portType name="TimeTrackerService">

<wsdl:operation name="createDailyReport">

<wsdl:input message="tns:createDailyReport" name="createDailyReport">

</wsdl:input>

<wsdl:output message="tns:createDailyReportResponse" name="createDailyReportResponse">

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="deleteUser">

<wsdl:input message="tns:deleteUser" name="deleteUser">

</wsdl:input>

<wsdl:output message="tns:deleteUserResponse" name="deleteUserResponse">

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="logoutUser">

<wsdl:input message="tns:logoutUser" name="logoutUser">

</wsdl:input>

<wsdl:output message="tns:logoutUserResponse" name="logoutUserResponse">

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="createUser">

<wsdl:input message="tns:createUser" name="createUser">

</wsdl:input>

<wsdl:output message="tns:createUserResponse" name="createUserResponse">

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="loginUser">

<wsdl:input message="tns:loginUser" name="loginUser">

</wsdl:input>

<wsdl:output message="tns:loginUserResponse" name="loginUserResponse">

</wsdl:output>

</wsdl:operation>

</wsdl:portType>

<wsdl:binding name="TimeTrackerServiceServiceSoapBinding" type="tns:TimeTrackerService">

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

<wsdl:operation name="createDailyReport">

<soap:operation soapAction="createDailyReport" style="document"/>

<wsdl:input name="createDailyReport">

<soap:body use="literal"/>

</wsdl:input>

<wsdl:output name="createDailyReportResponse">

<soap:body use="literal"/>

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="deleteUser">

<soap:operation soapAction="deleteUser" style="document"/>

<wsdl:input name="deleteUser">

<soap:body use="literal"/>

</wsdl:input>

<wsdl:output name="deleteUserResponse">

<soap:body use="literal"/>

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="logoutUser">

<soap:operation soapAction="logoutUser" style="document"/>

<wsdl:input name="logoutUser">

<soap:body use="literal"/>

</wsdl:input>

<wsdl:output name="logoutUserResponse">

<soap:body use="literal"/>

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="createUser">

<soap:operation soapAction="createUser" style="document"/>

<wsdl:input name="createUser">

<soap:body use="literal"/>

</wsdl:input>

<wsdl:output name="createUserResponse">

<soap:body use="literal"/>

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="loginUser">

<soap:operation soapAction="loginUser" style="document"/>

<wsdl:input name="loginUser">

<soap:body use="literal"/>

</wsdl:input>

<wsdl:output name="loginUserResponse">

<soap:body use="literal"/>

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name="TimeTrackerServiceService">

<wsdl:port binding="tns:TimeTrackerServiceServiceSoapBinding" name="TimeTrackerServicePort">

<soap:address location="http://localhost:8080/services/timeTrackerService"/>

</wsdl:port>

</wsdl:service>

</wsdl:definitions>

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Лист регистрации изменений** | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего  листов  (страниц)  в докум | №  документа | Входящий  № сопрово  дительного  документа  и дата | Подп. | Дата |
| Изм | изменен  ных | заме  ненных | новых | анулиро  ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |