

Современные ВЕБ-технологии

02.03.03 -Математическое обеспечение и администрирование информационных систем,
направленность (профиль) -разработка и администрирование информационных систем

<http://vikchas.ru>

Лекция 3. Тема «Современные технологии информационного взаимодействия и автоматизации»

Часовских Виктор Петрович
доктор технических наук, профессор кафедры
ШИиКМ, ФГБОУ ВО «Уральский государственный
экономический университет

Екатеринбург 2025

Современные технологии информационного взаимодействия и автоматизации

1. Современные стандарты информационного взаимодействия систем

REST API

REST (Representational State Transfer) - архитектурный стиль для распределенных систем.

Пример REST API запроса:

HTTP

GET /api/users/123 HTTP/1.1

Host: example.com

Accept: application/json

Пример ответа:

JSON

```
{  
  "id": 123,  
  "name": "Иван Петров",  
  "email": "ivan@example.com",  
  "created_at": "2023-09-15T14:30:00Z"  
}
```

Пример реализации REST API на Node.js/Express:

JAVASCRIPT

```
const express = require('express');  
const app = express();
```

```
app.get('/api/users/:id', (req, res) => {  
  const userId = req.params.id;
```

```
// Получение данных пользователя из БД
res.json({
  id: userId,
  name: "Иван Петров",
  email: "ivan@example.com",
  created_at: "2023-09-15T14:30:00Z"
});
});
```

```
app.listen(3000);
```

SOAP

SOAP (Simple Object Access Protocol) - протокол обмена структурированными сообщениями в XML.

Пример SOAP запроса:

XML

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
    <Authentication>
      <token>ABC123</token>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetUser>
      <userId>123</userId>
    </GetUser>
  </soap:Body>
</soap:Envelope>
```

GraphQL

GraphQL - язык запросов API и среда выполнения запросов.

Пример запроса GraphQL:

GraphQL

```
query {  
  user(id: "123") {  
    id  
    name  
    email  
    orders {  
      id  
      date  
      total  
    }  
  }  
}
```

WebSockets

Протокол для двусторонней связи через TCP-соединение.

JAVASCRIPT

// Серверная часть (Node.js с ws)

```
const WebSocket = require('ws');
```

```
const wss = new WebSocket.Server({ port: 8080 });
```

```
wss.on('connection', (ws) => {
```

```
  ws.on('message', (message) => {
```

```
    console.log(`Получено: ${message}`);
```

```
    ws.send(`Эхо: ${message}`);
```

```
  });
```

```
});
```

// Клиентская часть (JavaScript)

```
const socket = new WebSocket('ws://localhost:8080');
socket.onopen = () => {
  socket.send('Привет, сервер!');
};
socket.onmessage = (event) => {
  console.log(`Получено от сервера: ${event.data}`);
};
```

gRPC

Высокопроизводительный фреймворк RPC от Google.

Пример определения сервиса (Protocol Buffers):

PROTOBUF

syntax = "proto3";

```
service UserService {
  rpc GetUser (GetUserRequest) returns (User) {}
}
```

```
message GetUserRequest {
  int32 user_id = 1;
}
```

```
message User {
  int32 id = 1;
  string name = 2;
  string email = 3;
}
```

2. Программные средства и платформы инфраструктуры ИТ

Контейнеризация и оркестрация

Docker

DOCKERFILE

FROM node:14

WORKDIR /usr/src/app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 8080

CMD ["node", "server.js"]

Docker Compose

YAML

version: '3'

services:

app:

build: .

ports:

- "8080:8080"

depends_on:

- db

db:

image: postgres:13

environment:

POSTGRES_PASSWORD: example

POSTGRES_DB: myapp

Kubernetes

YAML

apiVersion: apps/v1

kind: Deployment

metadata:

name: myapp-deployment

spec:

replicas: 3

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec:

containers:

- name: myapp

image: myapp:1.0

ports:

- containerPort: 8080

Облачные платформы

- **IaaS**(Infrastructure as a Service): AWS EC2, Google Compute Engine, Microsoft Azure VMs
- **PaaS**(Platform as a Service): Heroku, Google App Engine, AWS Elastic Beanstalk
- **SaaS**(Software as a Service): Salesforce, Microsoft 365, Google Workspace

Системы мониторинга

- **Prometheus**: Сбор метрик
- **Grafana**: Визуализация метрик
- **ELK Stack**(Elasticsearch, Logstash, Kibana): Сбор и анализ логов

CI/CD

Пример конфигурации GitHub Actions:

YAML

name: CI/CD Pipeline

on:

push:

branches: [main]

jobs:

build-and-deploy:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Set up Node.js

uses: actions/setup-node@v2

with:

node-version: '14'

- name: Install dependencies

run: npm install

- name: Run tests

run: npm test

- name: Build

run: npm run build

- name: Deploy

uses: some-deployment-action@v1

with:

api-key: \${{ secrets.DEPLOY_API_KEY }}

3. Современные подходы и стандарты автоматизации организации

CRM (Customer Relationship Management)

Системы для управления взаимоотношениями с клиентами.

- **Salesforce**: Облачная CRM-платформа
- **Microsoft Dynamics 365**: Интегрированное решение для бизнеса
- **Битрикс24**: Российская CRM-система

Пример данных клиента в CRM:

JSON

```
{
  "customer_id": "C12345",
  "name": "ООО Прогресс",
  "status": "активный",
  "contacts": [
    {
      "type": "primary",
      "name": "Иванов Иван",
      "email": "ivanov@progress.ru",
      "phone": "+7 (999) 123-45-67"
    }
  ],
  "deals": [
    {
      "id": "D001",
      "title": "Поставка оборудования",
      "value": 450000,
      "stage": "в работе",
      "probability": 80
    }
  ]
}
```

```
}  
]  
}
```

ERP (Enterprise Resource Planning)

Системы для управления ресурсами предприятия.

- **SAP S/4HANA:** Комплексная ERP-система от SAP
- **Oracle ERP Cloud:** Облачное ERP-решение
- **1C:ERP:** Российское решение для управления предприятием

Ключевые модули ERP:

- Финансы и бухгалтерия
- Управление персоналом
- Управление запасами
- Управление производством
- Управление цепочками поставок

MRP (Material Requirements Planning)

Системы планирования потребности в материалах.

Пример расчета MRP:

Продукт А требует:

- 2 единицы компонента В
- 3 единицы компонента С

План производства:

- Неделя 1: 100 единиц продукта А
- Неделя 2: 150 единиц продукта А

Потребность в компонентах:

- Неделя 1: 200 единиц В, 300 единиц С
- Неделя 2: 300 единиц В, 450 единиц С

ITIL (IT Infrastructure Library)

Библиотека инфраструктуры информационных технологий - сборник лучших практик управления ИТ-услугами.

Ключевые процессы ITIL:

1. Управление инцидентами
2. Управление проблемами
3. Управление изменениями
4. Управление доступностью
5. Управление уровнем услуг

ITSM (IT Service Management)

Подход к управлению ИТ-услугами, ориентированный на потребности бизнеса.

Пример жизненного цикла инцидента в ITSM:

1. Регистрация инцидента
2. Категоризация и приоритезация
3. Первичная диагностика
4. Эскалация (при необходимости)
5. Расследование и диагностика
6. Решение и восстановление
7. Закрытие инцидента

4. Системы классификации и кодирования информации

Классификаторы и справочники

- **ОКПД 2**(Общероссийский классификатор продукции по видам экономической деятельности)
- **ОКВЭД 2**(Общероссийский классификатор видов экономической деятельности)
- **ISO 3166**(Коды стран)
- **ISO 4217**(Коды валют)
- **SWIFT/BIC**(Коды банков)

Пример кода ОКВЭД:

62.01 - Разработка компьютерного программного обеспечения

Штриховое кодирование

- **EAN-13:** Европейский стандарт для товаров
- **Code 128:** Универсальный для буквенно-цифровых данных
- **QR-код:** Двумерный код для хранения информации

Пример генерации QR-кода на JavaScript:

JAVASCRIPT

```
const QRCode = require('qrcode');

QRCode.toFile('qrcode.png', 'https://example.com', {
  errorCorrectionLevel: 'H',
  margin: 1,
  width: 300
}, function(err) {
  if (err) throw err;
  console.log('QR код создан!');
});
```

Системы идентификации документов

Пример структуры номера документа:

ТН-2023/09-00123

| | └─── Порядковый номер
| └─── Месяц/год
└─── Тип документа (товарная накладная)

Реализация генератора номеров документов на Python:

PYTHON

```
import datetime

class DocumentNumberGenerator:
    def __init__(self):
        self.counters = {}
```

```

def generate_number(self, doc_type):
    today = datetime.datetime.now()
    month_year = today.strftime("%m/%y")
    period_key = f"{doc_type}-{month_year}"

    if period_key not in self.counters:
        self.counters[period_key] = 0

    self.counters[period_key] += 1
    counter = self.counters[period_key]

    return f"{doc_type}-20{month_year}-{counter:05d}"

```

Пример использования

```

generator = DocumentNumberGenerator()
invoice_number = generator.generate_number("ИН")
print(invoice_number) # ИН-2023/09-00001

```

Классификация элементов справочников

Пример иерархической структуры справочника:

JSON

```

{
  "name": "Организационная структура",
  "items": [
    {
      "code": "01",
      "name": "Головной офис",
      "items": [
        {

```

```
"code": "01.01",
"name": "Административный отдел"
},
{
  "code": "01.02",
  "name": "ИТ-отдел",
  "items": [
    {
      "code": "01.02.01",
      "name": "Отдел разработки"
    },
    {
      "code": "01.02.02",
      "name": "Отдел инфраструктуры"
    }
  ]
}
],
{
  "code": "02",
  "name": "Филиал в Санкт-Петербурге"
}
]
```

Эти стандарты, подходы и технологии обеспечивают эффективное взаимодействие информационных систем и процессов в современной организации.