web3j is a lightweight, reactive, type safe Java and Android library for integrating with clients (nodes) on the Ethereum network.

This allows you to work with the Ethereum blockchain, without the additional overhead of having to write your own integration code for the platform.

The Java and the Blockchain talk provides an overview of blockchain, Ethereum and web3j.
Eclipse Scout
What is Eclipse Scout?

Business Application Framework

- Open Source Eclipse Project
- Based on Java and HTML5
- Multi Device support, Modular Apps, ...

Goals

- Long Term Sustainability (enterprise apps live > 10 years)
- Boosts Productivity (producing software in Switzerland ...)
- Easy to learn (newbies productive in 1-2 weeks)
Eclipse Scout The Java Story

Application Model

- (Very) Long Term
- Clean Business Code
- High Maintainability

```java
@Order(10)
public class FirstNameField extends AbstractStringField {
    @Override
    protected String getConfigurerLabel() {
        return TEXTS.get("FirstName");
    }
}
```
Eclipse Scout The HTML5 Story

Rendering

- HTML5, CSS3, JavaScript
- Styling & theming
Eclipse Scout Commercial Application
Eclipse Scout Hello World

Create the Scout "Hello World" application
Create a new Scout maven-based archetype project

Import the Scout demo applications
Import the Scout demo apps using Eclipse Oomph. In the installer search for "scout" and tick "Scout Demo App".

Checkout projects from Git
Checkout Eclipse projects hosted in a Git repository

Review IDE configuration settings
Review the IDE's most fiercely contested preferences

Message from Server
Message: Hello Eclipse DemoCamp!
Server Time: 05/29/2017 2:10 PM
Machine Learning
Deeplearning4j
Deep Learning Library

→ Open Source
→ **Java** (most are Python)
→ Good documentation

Features

→ Full GPU support
→ Distributed deep learning
→ Runs with Hadoop + Spark

https://github.com/deeplearning4j/deeplearning4j
The ML «Hello World» Recognition of handwritten digits

PROC. OF THE IEEE, NOVEMBER 1998

Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network whose weights are constrained to be identical between C1 and C2.

1998 Gradient-based Learning for Document Recognition, Y. LeCun

```java
public static MultilayerConfiguration configuration() {
    return new NeuralNetConfiguration.Builder()
        .seed(SEED).weightInit(WeightInit.XAVER)
        .iterations(NUM_ITERATIONS)
        .regularization(true).l2(0.0005).learningRate(.01)
        .optimizationAlgorithm(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
        .updater(Updater.NESTEROVS).momentum(0.9)
        .list()
        .layer(0, new ConvolutionLayer.Builder(5, 5)
            .stride(1, 1)
            .nIn(NUM_CHANNELS)
            .nOut(26)
            .activation(Activation.IDENTITY)
            .build())
        .layer(1, new SubsamplingLayer.Builder(SubsamplingLayer.PoolingType.MAX)
            .kernelSize(2, 2)
            .stride(2, 2)
            .build())
        .layer(2, new ConvolutionLayer.Builder(5, 5).stride(1, 1)
            .nOut(50)
            .activation(Activation.IDENTITY)
            .build())
        .layer(3, new SubsamplingLayer.Builder(SubsamplingLayer.PoolingType.MAX)
            .kernelSize(2, 2)
            .stride(2, 2)
            .build())
        .layer(4, new DenseLayer.Builder()
            .activation(Activation.RELU)
            .nOut(500)
            .build())
        .layer(5, new OutputLayer.Builder(lossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
            .activation(Activation.SOFTMAX)
            .nOut(NUM_OUTPUTS)
            .build())
        .setInputType(InputType.convolutionalFlat(28, 28, 1))
        .backprop(true)
        .pretrain(false).build();
```
Handwritten Data
Training of a Neural Network Model

```java
/**
 * Train the network for the specified number of epochs.
 */
public void train(DataSetIterator trainData, DataSetIterator validationData, int epochs) {
    for(int epoch = 1; epoch <= epochs; epoch++) {

        // train the network using training data
        log.info("Starting epoch {}, samples: {}", epoch, trainData.numExamples());
        trainData.reset();
        m_network.fit(trainData);

        // evaluate performance using validation data
        validationData.reset();
        evaluate(validationData);
    }
}
```
Error Rate and Training Epochs

Numbers (BSI Data)
9’000 Training images (18 writers)
1’000 Test images (2 writers)
Error Rate and Training Epochs

Numbers + MNIST
Start with MNIST Modell
Train on Numbers
40% Error reduction
Blockchain
web3j/Ethereum
Blockchain «Micro-Intro»

Blockchain

- **Bitcoin** started the field in 2009
- Main Features: Cheap, fast, efficient (traditional setup: T+3 and high fees)
- **Ethereum** adds smart contracts
- Remarkable achievements and much hype

Main Challenges

- Scalability
- Privacy
- Regulatory & legal
Ethereum Smart Contracts

What is it?

- Piece of byte code (usually written in Solidity)
- Is executed by the Ethereum Virtual Machine (EVM)
- Has an owner

Examples

- Greeter (the Ethereum «Hello World»)
- «Truly» autonomous cars
contract greeter {

    /* Owner of this contract */
    address owner;

    /* Configurable greeting */
    string greeting;

    /* Constructor runs when contract is deployed */
    function greeter(string _greeting) public {
        owner = msg.sender;
        greeting = _greeting;
    }

    /* Main function */
    function greet() constant returns (string) {
        return greeting;
    }

    /* Function to recover the funds on the contract */
    function kill() {
        if (msg.sender == owner)
            selfdestruct(owner);
    }
}
«Truly» Autonomous Cars

Uber’s self-driving cars are now picking up passengers in Arizona

Tempe or bust!
by Andrew J. Haeftens | @andryhafink | Feb 21, 2017, 1:33pm EST

Smart contract: To order car to transport people (by paying to contract)
Smart contract: Car pays for energy/services

A subsidiary of RWE, one of Germany’s biggest energy and gas provider with 30 million customers and billions of revenue, has launched 100s of electronic vehicles (EV) charging stations all over Germany, connected to ethereum’s public blockchain.
Ethereum and Application Integration

- **Ethereum Client**
  - Geth
  - TestRPC

- **Ethereum Peer-to-Peer Network**
  - Interface
    - JSON-RPC
  - http://localhost:8545

- **JavaScript**
  - web3

- **Java**
  - web3j
web3j

Library to interact with Ethereum (its peer-to-peer clients)
- Open Source
- Java (default is JavaScript)
- Good documentation

Features
- JSON-RPC client API implementation
- Tool to generate Contract Wrappers in Java

https://github.com/web3j/web3j
Web3j: Generated Contract Wrapper

```java
public final class Greeter extends Contract {
    private Greeter(String contractAddress, Web3j web3j, Credentials credentials, BigInteger gasPrice, BigInteger gasLimit) {
        super(contractAddress, web3j, credentials, gasPrice, gasLimit);
    }

    private Greeter(String contractAddress, Web3j web3j, TransactionManager transactionManager, BigInteger gasPrice, BigInteger gasLimit) {
        super(contractAddress, web3j, transactionManager, gasPrice, gasLimit);
    }

    public Future<Uint256> deposits() {
        Function function = new Function("deposits",
            Arrays.<Type>asList(),
            Arrays.<TypeReference<?>>asList(new TypeReference<Uint256>() {}));
        return executeCallSingleValueReturnAsync(function);
    }
}
```
Thanks!

@ZimMatthias