

HDFS API

File System Java API

- `org.apache.hadoop.fs.FileSystem`
 - Абстрактный класс, которые представляет абстрактную файловую систему
 - (!) Это именно класс, а не интерфейс
- Реализуется в различных вариантах

Реализации *FileSystem*

Hadoop предоставляет несколько конкретных реализаций:

- *org.apache.hadoop.fs.LocalFileSystem*

- Подходит для нативных FS, использующих локальные диски

- *org.apache.hadoop.hdfs.DistributedFileSystem*

- Hadoop Distributed File System (HDFS)

- *org.apache.hadoop.hdfs.HftpFileSystem*

- Доступ к HDFS в read-only режиме через HTTP

- *org.apache.hadoop.fs.ftp.FTPFileSystem*

- Файловая система поверх FTP-сервера

```
public class SimpleLocals {  
    public static void main(String[] args) throws Exception{  
  
        Path path = new Path("/");  
        if ( args.length == 1 ){  
            path = new Path(args[0]);  
        }  
  
        Configuration conf = new Configuration();  
        FileSystem fs = FileSystem.get(conf);  
  
        FileStatus [] files = fs.listStatus(path);  
        for (FileStatus file : files ){  
            System.out.println(file.getPath().getName());  
        }  
    }  
}
```

FileSystem API: Path

- Объект Path представляет файл или директорию
 - *java.io.File* сильно завязан на локальную FS
- Path – это на самом деле URI в FS
 - HDFS: *hdfs://localhost/user/file1*
 - Local: *file:///user/file1*

```
new Path("/test/file1.txt");  
new Path("hdfs://localhost:9000/test/");
```

Объект Configuration

- Объект Configuration хранит конфигурацию сервера и клиента
- Использует простую парадигму key-value
- Получения значения:

```
String name = conf.get("fs.default.name");
```

```
String name = conf.get("fs.default.name", "hdfs://localhost:9000");
```

```
float size = conf.getFloat("file.size");
```

Чтение данных из файла

- Создать объект *FileSystem*
- Открыть *InputStream*, указывающий на *Path*
- Скопировать данные по байтам используя *IOUtils*
- Заккрыть *InputStream*

```
public class ReadFile {  
    public static void main(String[] args) throws  
                                IOException {  
        Path file = new Path("/path/to/file.txt");  
        FileSystem fs = FileSystem.get(new  
            Configuration());  
  
        InputStream input = null;  
        try {  
            input = fs.open(file);  
            IOUtils.copyBytes(input, System.out, 4096);  
        } finally {  
            IOUtils.closeStream(input);  
        }  
    }  
}
```


Запись данных в файл

- Создать объект *FileSystem*
- Открыть *OutputStream*
 - Указывает на *Path* из *FileSystem*
 - Используем *FSDDataOutputStream*
 - Автоматически создаются все директори в пути, если не существуют
- Копируем данные по байтам используя *IOUtils*

```
public class WriteToFile {  
    public static void main(String[] args) throws  
        IOException {  
        String text = "Hello world in HDFS!\n";  
        InputStream in = new BufferedInputStream(  
            new ByteArrayInputStream(text.getBytes()));  
  
        Path file = new Path("/path/to/file.txt");  
        Configuration conf = new Configuration();  
  
        FileSystem fs = FileSystem.get(conf);  
  
        FSDataOutputStream out = fs.create(file);  
        IOUtils.copyBytes(in, out, conf); // Copy Data  
    }  
}
```

FileSystem: запись данных

- ***fs.append(path)*** – дописать к существующему файлу
 - Поддержка для HDFS
- Нельзя записать в середину файла
- ***FileSystem.create(Path)*** создает все промежуточные директории для заданного каталога (по умолчанию)
 - Если это не нужно, то надо использовать
 - *public FSDataOutputStream create(Path f, boolean overwrite)*
 - *overwrite = false*

FileSystem: подстановки (globbing)

- **FileSystem** имеет поддержку матчинга имени файла по заданному паттерну используя метод ***globStatus()***
 - *FileStatus [] files = fs.globStatus(glob);*
- Примеры шаблонов
 - ? – любой один символ
 - * - любые 0 и больше символов
 - [abc] – любой символ из набора в скобках
 - [^a] – любой символ, кроме указанного
 - {ab,cd} – любая строка из указанных в скобках