

## src

# isphere\_exceptions package

## Submodules

### isphere\_exceptions.auto\_browser module

*exception* **isphere\_exceptions.auto\_browser.AutoBrowserErrorInterface**(\*args, \*\*kwargs)

Базовые классы: `ISphereException`

Интерфейс ошибки управляемого браузера

Ошибка с кодом 525, префикс BRW.

**EXCEPTION\_PREFIX= 'BRW'**

Дефолтный префикс для внутреннего кода ошибки

**TEMPLATE\_CONTEXT:** `Optional [ str ] = 'automated browser'`

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

*exception* **isphere\_exceptions.auto\_browser.AutoBrowserError**(\*args, \*\*kwargs)

Базовые классы: `AutoBrowserErrorInterface`, `CommonException`

*exception* **isphere\_exceptions.auto\_browser.AutoBrowserConfigurationInvalid**(\*args, \*\*kwargs)

Базовые классы: `AutoBrowserErrorInterface`, `ConfigurationInvalidException`

*exception* **isphere\_exceptions.auto\_browser.AutoBrowserConnection**(\*args, \*\*kwargs)

Базовые классы: `AutoBrowserErrorInterface`, `ConnectionException`

*exception* **isphere\_exceptions.auto\_browser.AutoBrowserTimeout**(\*args, \*\*kwargs)

Базовые классы: `AutoBrowserErrorInterface`, `TimeoutException`

**exception** `isphere_exceptions.auto_browser.AutoBrowserOperationFailure(*args, **kwargs)`

Базовые классы: `AutoBrowserErrorInterface`, `OperationFailureException`

**exception** `isphere_exceptions.auto_browser.AutoBrowserParseError(*args, **kwargs)`

Базовые классы: `AutoBrowserErrorInterface`, `ParseExceptionException`

## `isphere_exceptions.ja3` module

**exception** `isphere_exceptions.ja3.JA3ErrorInterface(*args, **kwargs)`

Базовые классы: `ISphereException`

Интерфейс ошибки для работы с ja3 server (tls proxy)

Ошибка с кодом 524, префикс JA3.

**EXCEPTION\_PREFIX= 'JA3'**

Дефолтный префикс для внутреннего кода ошибки

**TEMPLATE\_CONTEXT:** `Optional [ str ] = 'JA3'`

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**exception** `isphere_exceptions.ja3.JA3Error(*args, **kwargs)`

Базовые классы: `JA3ErrorInterface`, `CommonException`

**exception** `isphere_exceptions.ja3.JA3ServerConfigurationInvalid(*args, **kwargs)`

Базовые классы: `JA3ErrorInterface`, `ConfigurationInvalidException`

**exception** `isphere_exceptions.ja3.JA3ServerConnection(*args, **kwargs)`

Базовые классы: `JA3ErrorInterface`, `ConnectionException`

**exception** `isphere_exceptions.ja3.JA3ServerTimeout(*args, **kwargs)`

Базовые классы: `JA3ErrorInterface`, `TimeoutException`

**exception** `isphere_exceptions.ja3.JA3ServerOperationFailure(*args, **kwargs)`

Базовые классы: `JA3ErrorInterface`, `OperationFailureException`

**exception** `isphere_exceptions.ja3.JA3ServerParseError(*args, **kwargs)`

Базовые классы: `JA3ErrorInterface` , `ParseException`

**exception** `isphere_exceptions.ja3.JA3Locked(*args, **kwargs)`

Базовые классы: `JA3ErrorInterface` , `LockedException`

**exception** `isphere_exceptions.ja3.JA3Blocked(*args, **kwargs)`

Базовые классы: `JA3ErrorInterface` , `BlockedException`

**exception** `isphere_exceptions.ja3.JA3VersionTLS(*args, internal_code=508, **kwargs)`

Базовые классы: `JA3ErrorInterface` , `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE**= *'Данная версия TLS {} для источника не поддерживается'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**livenessprobe**= *False*

**log\_level**= *'error'*

**exception** `isphere_exceptions.ja3.JA3InvalidPayload(*args, internal_code=509, **kwargs)`

Базовые классы: `JA3ErrorInterface` , `ErrorReturnToQueue`

**DEFAULT\_MESSAGE**= *'Неверное тело запроса при отправке запроса {}'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception** `isphere_exceptions.ja3.JA3MismatchUserAgent(*args, internal_code=510, **kwargs)`

Базовые классы: `JA3ErrorInterface` , `ErrorReturnToQueue`

**DEFAULT\_MESSAGE**  
= *'Данный {} не может использоваться в связке с данным User-Agent. Проверьте введенные данные'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception** `isphere_exceptions.ja3.JA3HandshakeError(*args, internal_code=511, **kwargs)`

Базовые классы: `JA3ErrorInterface` , `ErrorReturnToQueue`

**DEFAULT\_MESSAGE**= *'Ошибка подключения {}'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

## isphere\_exceptions.keydb module

**exception** isphere\_exceptions.keydb.KeyDBErrorInterface(\*args, \*\*kwargs)

Базовые классы: `ISphereException`

Интерфейс ошибки для работы с KeyDB

Ошибка с кодом 520, префикс KDB.

**EXCEPTION\_PREFIX= 'KDB'**

Дефолтный префикс для внутреннего кода ошибки

**TEMPLATE\_CONTEXT:** `Optional [ str ] = 'KeyDB'`

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**exception** isphere\_exceptions.keydb.KeyDBError(\*args, \*\*kwargs)

Базовые классы: `KeyDBErrorInterface`, `CommonException`

**exception** isphere\_exceptions.keydb.KeyDBConfigurationInvalid(\*args, \*\*kwargs)

Базовые классы: `KeyDBErrorInterface`, `ConfigurationInvalidException`

**exception** isphere\_exceptions.keydb.KeyDBConnection(\*args, \*\*kwargs)

Базовые классы: `KeyDBErrorInterface`, `ConnectionException`

**exception** isphere\_exceptions.keydb.KeyDBTimeout(\*args, \*\*kwargs)

Базовые классы: `KeyDBErrorInterface`, `TimeoutException`

**exception** isphere\_exceptions.keydb.KeyDBOperationFailure(\*args, \*\*kwargs)

Базовые классы: `KeyDBErrorInterface`, `OperationFailureException`

**exception** isphere\_exceptions.keydb.KeyDBParseError(\*args, \*\*kwargs)

Базовые классы: `KeyDBErrorInterface`, `ParseErrorException`

## isphere\_exceptions.mongo module

**exception** isphere\_exceptions.mongo.MongoErrorInterface(\*args, \*\*kwargs)

Базовые классы: `ISphereException`

Интерфейс ошибки для работы с MongoDB

Ошибка с кодом 522, префикс MNG.

**EXCEPTION\_PREFIX= 'MNG'**

Дефолтный префикс для внутреннего кода ошибки

**TEMPLATE\_CONTEXT:** `Optional [ str ] = 'MongoDB'`

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**exception isphere\_exceptions.mongo.MongoError(\*args, \*\*kwargs)**

Базовые классы: `MongoErrorInterface`, `CommonException`

**exception isphere\_exceptions.mongo.MongoConfigurationInvalid(\*args, \*\*kwargs)**

Базовые классы: `MongoErrorInterface`, `ConfigurationInvalidException`

**exception isphere\_exceptions.mongo.MongoConnection(\*args, \*\*kwargs)**

Базовые классы: `MongoErrorInterface`, `ConnectionException`

**exception isphere\_exceptions.mongo.MongoTimeout(\*args, \*\*kwargs)**

Базовые классы: `MongoErrorInterface`, `TimeoutException`

**exception isphere\_exceptions.mongo.MongoOperationFailure(\*args, \*\*kwargs)**

Базовые классы: `MongoErrorInterface`, `OperationFailureException`

**exception isphere\_exceptions.mongo.MongoParseError(\*args, \*\*kwargs)**

Базовые классы: `MongoErrorInterface`, `ParseErrorException`

## isphere\_exceptions.proxy module

**exception isphere\_exceptions.proxy.ProxyErrorInterface(\*args, \*\*kwargs)**

Базовые классы: `ISphereException`

Интерфейс ошибки для работы с прокси

Ошибка с кодом 521, префикс PRX.

**EXCEPTION\_PREFIX= 'PRX'**

Дефолтный префикс для внутреннего кода ошибки

**TEMPLATE\_CONTEXT:** Optional [ str ] = 'proxy'

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**exception** `isphere_exceptions.proxy.ProxyError(*args, **kwargs)`

Базовые классы: `ProxyErrorInterface`, `CommonException`

**exception** `isphere_exceptions.proxy.ProxyServerConfigurationInvalid(*args, **kwargs)`

Базовые классы: `ProxyErrorInterface`, `ConfigurationInvalidException`

**TEMPLATE\_CONTEXT:** Optional [ str ] = 'сервису proxy'

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**exception** `isphere_exceptions.proxy.ProxyServerConnection(*args, **kwargs)`

Базовые классы: `ProxyErrorInterface`, `ConnectionException`

**TEMPLATE\_CONTEXT:** Optional [ str ] = 'сервису proxy'

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**exception** `isphere_exceptions.proxy.ProxyServerTimeout(*args, **kwargs)`

Базовые классы: `ProxyErrorInterface`, `TimeoutException`

**TEMPLATE\_CONTEXT:** Optional [ str ] = 'сервису proxy'

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**exception** `isphere_exceptions.proxy.ProxyServerOperationFailure(*args, **kwargs)`

Базовые классы: `ProxyErrorInterface`, `OperationFailureException`

**TEMPLATE\_CONTEXT:** Optional [ str ] = 'сервису proxy'

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**exception** `isphere_exceptions.proxy.ProxyServerParseError(*args, **kwargs)`

Базовые классы: `ProxyErrorInterface`, `ParseErrorException`

**exception** `isphere_exceptions.proxy.ProxyLocked(*args, **kwargs)`

Базовые классы: `ProxyErrorInterface` , `LockedException`

*exception* `isphere_exceptions.proxy.ProxyBlocked(*args, **kwargs)`

Базовые классы: `ProxyErrorInterface` , `BlockedException`

*exception* `isphere_exceptions.proxy.ProxyTimeout(*args, internal_code=508, **kwargs)`

Базовые классы: `ProxyErrorInterface` , `ErrorReturnToQueue`

**DEFAULT\_MESSAGE=** *'Превышен таймаут ожидания ответа от {}. Возможна ротация IP'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

*exception* `isphere_exceptions.proxy.ProxyAuthenticationRequired(*args, internal_code=509, **kwargs)`

Базовые классы: `ProxyErrorInterface` , `ErrorReturnToQueue`

**DEFAULT\_MESSAGE=** *'Ошибка в авторизации {}. Проверьте логин и пароль'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

*exception* `isphere_exceptions.proxy.ProxyUnavailable(*args, internal_code=510, **kwargs)`

Базовые классы: `ProxyErrorInterface` , `ErrorReturnToQueue`

**DEFAULT\_MESSAGE=** *'Недоступен {}'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

*exception* `isphere_exceptions.proxy.ProxyTemporaryUnavailable(*args, internal_code=511, **kwargs)`

Базовые классы: `ProxyErrorInterface` , `ErrorReturnToQueue`

**DEFAULT\_MESSAGE=** *'Временно недоступен {}'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

## **isphere\_exceptions.rabbitmq module**

*exception* `isphere_exceptions.rabbitmq.RabbitMQErrorInterface(*args, **kwargs)`

Базовые классы: `ISphereException`

Интерфейс ошибки для работы с RabbitMQ

Ошибка с кодом 523, префикс RAB

```
EXCEPTION_PREFIX= 'RAB'
```

Дефолтный префикс для внутреннего кода ошибки

```
TEMPLATE_CONTEXT: Optional [ str ] = 'RabbitMQ'
```

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

```
exception isphere_exceptions.rabbitmq.RabbitMQError(*args, **kwargs)
```

Базовые классы: `RabbitMQErrorInterface`, `CommonException`

```
exception isphere_exceptions.rabbitmq.RabbitMQConfigurationInvalid(*args, **kwargs)
```

Базовые классы: `RabbitMQErrorInterface`, `ConfigurationInvalidException`

```
exception isphere_exceptions.rabbitmq.RabbitMQConnection(*args, **kwargs)
```

Базовые классы: `RabbitMQErrorInterface`, `ConnectionException`

```
exception isphere_exceptions.rabbitmq.RabbitMQTimeout(*args, **kwargs)
```

Базовые классы: `RabbitMQErrorInterface`, `TimeoutException`

```
exception isphere_exceptions.rabbitmq.RabbitMQOperationFailure(*args, **kwargs)
```

Базовые классы: `RabbitMQErrorInterface`, `OperationFailureException`

```
exception isphere_exceptions.rabbitmq.RabbitMQParseError(*args, **kwargs)
```

Базовые классы: `RabbitMQErrorInterface`, `ParseErrorException`

## isphere\_exceptions.session module

```
exception isphere_exceptions.session.SessionErrorInterface(*args, code=526, **kwargs)
```

Базовые классы: `ISphereException`

Интерфейс ошибки для работы с сессиями

Ошибка с кодом 526, префикс SSS

```
EXCEPTION_PREFIX= 'SSS'
```

Дефолтный префикс для внутреннего кода ошибки

```
TEMPLATE_CONTEXT: Optional [ str ] = 'Сессия'
```



Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**exception** `isphere_exceptions.session.SessionError(*args, code=526, **kwargs)`

Базовые классы: `SessionErrorInterface`, `CommonException`

**DEFAULT\_MESSAGE**= *'Ошибка использований сессии'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception** `isphere_exceptions.session.SessionOutdated(*args, internal_code=501, **kwargs)`

Базовые классы: `SessionErrorInterface`, `ErrorReturnToQueue`

**DEFAULT\_MESSAGE**= *'Истек срок жизни сессии'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception** `isphere_exceptions.session.SessionInvalidCredentials(*args, internal_code=502, **kwargs)`

Базовые классы: `SessionErrorInterface`, `ErrorReturnToQueue`

**DEFAULT\_MESSAGE**= *'Неверные авторизационные данные'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception** `isphere_exceptions.session.SessionEmpty(*args, internal_code=503, code=512, **kwargs)`

Базовые классы: `SessionErrorInterface`, `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE**= *'Недостаток сессий'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**livenessprobe**= *True*

**log\_level**= *'error'*

**exception** `isphere_exceptions.session.SessionCaptchaDecodeError(*args, internal_code=504, **kwargs)`

Базовые классы: `SessionErrorInterface`, `ErrorReturnToQueue`

**DEFAULT\_MESSAGE**= *'Возникла ошибка расшифровки капчи'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
exception isphere_exceptions.session.SessionCaptchaDetected(*args,  
internal_code=505, **kwargs)
```

Базовые классы: `SessionErrorInterface` , `ErrorReturnToQueue`

```
DEFAULT_MESSAGE= 'Обнаружена капча во время использования сессии'
```

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
exception isphere_exceptions.session.SessionLocked(*args, code=526, **kwargs)
```

Базовые классы: `SessionErrorInterface` , `LockedException`

```
exception isphere_exceptions.session.SessionBlocked(*args, code=526, **kwargs)
```

Базовые классы: `SessionErrorInterface` , `BlockedException`

```
exception isphere_exceptions.session.SessionLimitError(*args, internal_code=508,  
**kwargs)
```

Базовые классы: `SessionErrorInterface` , `ErrorReturnToQueue`

```
DEFAULT_MESSAGE= 'Превышен лимит использования сессии'
```

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

## isphere\_exceptions.source module

```
exception isphere_exceptions.source.SourceErrorInterface(*args, code=530, **kwargs)
```

Базовые классы: `ISphereException`

Интерфейс ошибки для работы с источником (сайт, приложение)

Ошибка с кодом 530, префикс SRC

```
EXCEPTION_PREFIX= 'SRC'
```

Дефолтный префикс для внутреннего кода ошибки

```
TEMPLATE_CONTEXT: Optional [ str ] = 'источник'
```

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

```
exception isphere_exceptions.source.SourceError(*args, code=530, **kwargs)
```

Базовые классы: `SourceErrorInterface` , `CommonException`

**DEFAULT\_MESSAGE=** *'Ошибка со стороны источника'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

*exception* `isphere_exceptions.source.SourceParseError(*args, internal_code=501, **kwargs)`

Базовые классы: `SourceErrorInterface` , `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE=** *'Неподдерживаемый ответ источника'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**livenessprobe=** *False*

**log\_level=** *'warning'*

*exception* `isphere_exceptions.source.SourceConnection(*args, internal_code=502, **kwargs)`

Базовые классы: `SourceErrorInterface` , `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE=** *'Источник не отвечает на запросы'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**livenessprobe=** *True*

*exception* `isphere_exceptions.source.SourceDown(*args, internal_code=503, **kwargs)`

Базовые классы: `SourceErrorInterface` , `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE**  
**=** *'Источник выключен на стороне. Возможно сайт/приложение недоступно'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**livenessprobe=** *True*

*exception* `isphere_exceptions.source.SourceTimeout(*args, internal_code=504, **kwargs)`

Базовые классы: `SourceErrorInterface` , `ErrorReturnToQueue`

**DEFAULT\_MESSAGE=** *'Превышен таймаут запроса к источнику'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
exception isphere_exceptions.source.SourceIncorrectDataDetected(*args,  
internal_code=505, **kwargs)
```

Базовые классы: `SourceErrorInterface` , `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE=** *'Источник не может выполнить запрос по указанным данным'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**log\_level=** *'warning'*

**livenessprobe=** *True*

```
exception isphere_exceptions.source.SourceVagueData(*args, internal_code=506,  
**kwargs)
```

Базовые классы: `SourceErrorInterface` , `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE=** *'Найдено слишком много совпадений'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**log\_level=** *'warning'*

**livenessprobe=** *True*

```
exception isphere_exceptions.source.SourceLimitError(*args, internal_code=507,  
**kwargs)
```

Базовые классы: `SourceErrorInterface` , `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE=** *'Превышен лимит использования источника'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
exception isphere_exceptions.source.SourceConfigurationInvalid(*args,  
internal_code=509, **kwargs)
```

Базовые классы: `SourceErrorInterface` , `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE**  
= *'Заданная конфигурация использования источника некорректна. Проверьте введенные параметры и на'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception** `isphere_exceptions.source.SourceOperationFailure(*args, internal_code=509, **kwargs)`

Базовые классы: `SourceErrorInterface` , `ErrorReturnToQueue`

**DEFAULT\_MESSAGE**= 'При выполнении операции возникла ошибка'

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

## `isphere_exceptions.sql_db module`

**exception** `isphere_exceptions.sql_db.SQLDBErrorInterface(*args, **kwargs)`

Базовые классы: `ISphereException`

Интерфейс ошибки для работы с базами SQL

Ошибка с кодом 527, префикс SQL

**EXCEPTION\_PREFIX**= 'SQL'

Дефолтный префикс для внутреннего кода ошибки

**TEMPLATE\_CONTEXT**: `Optional [ str ]` = 'SQL DB'

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**exception** `isphere_exceptions.sql_db.SQLDBError(*args, **kwargs)`

Базовые классы: `SQLDBErrorInterface` , `CommonException`

**exception** `isphere_exceptions.sql_db.SQLDBConfigurationInvalid(*args, **kwargs)`

Базовые классы: `SQLDBErrorInterface` , `ConfigurationInvalidException`

**exception** `isphere_exceptions.sql_db.SQLDBConnection(*args, **kwargs)`

Базовые классы: `SQLDBErrorInterface` , `ConnectionException`

**exception** `isphere_exceptions.sql_db.SQLDBTimeout(*args, **kwargs)`

Базовые классы: `SQLDBErrorInterface` , `TimeoutException`

**exception** `isphere_exceptions.sql_db.SQLDBOperationFailure(*args, **kwargs)`

Базовые классы: `SQLDBErrorInterface` , `OperationFailureException`

**exception** `isphere_exceptions.sql_db.SQLDBParseError(*args, **kwargs)`

Базовые классы: `SQLDBErrorInterface` , `ParseErrorException`

## isphere\_exceptions.success module

**exception** isphere\_exceptions.success.FoundEvent(\*args, internal\_code=200, \*\*kwargs)

Базовые классы: `ErrorNoReturnToQueue`

Событие найденных данных в источнике

Код 200, префикс SCS

**DEFAULT\_MESSAGE**= 'Найден'

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception** isphere\_exceptions.success.CachedEvent(\*args, internal\_code=201, \*\*kwargs)

Базовые классы: `ErrorNoReturnToQueue`

Событие найденных данных в источнике, использован кеш ответа

Код 201, префикс SCS

**DEFAULT\_MESSAGE**= 'Найден, использован кеш предыдущего ответа'

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception** isphere\_exceptions.success.NoDataEvent(\*args, internal\_code=204, \*\*kwargs)

Базовые классы: `ErrorNoReturnToQueue`

Событие не найденных данных в источнике

Код 204, префикс SCS

**DEFAULT\_MESSAGE**= 'Не найден'

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception** isphere\_exceptions.success.PartialContentEvent(\*args, internal\_code=206, \*\*kwargs)

Базовые классы: `ErrorNoReturnToQueue`

Событие частично найденных данных в источнике

Код 206, префикс SCS

**DEFAULT\_MESSAGE**= 'Частично найден'

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

## isphere\_exceptions.worker module

```
exception isphere_exceptions.worker.InternalWorkerErrorInterface(*args,  
code=599, **kwargs)
```

Базовые классы: `ISphereException`

Интерфейс ошибки обработчика

Ошибка с кодом 500-599, префикс INT

**EXCEPTION\_PREFIX**= 'INT'

Дефолтный префикс для внутреннего кода ошибки

**TEMPLATE\_CONTEXT**: `Optional [ str ]` = 'Обработчик'

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

```
exception isphere_exceptions.worker.InternalWorkerError(*args, code=599, **kwargs)
```

Базовые классы: `InternalWorkerErrorInterface`, `CommonException`

**DEFAULT\_MESSAGE**= 'Внутренняя ошибка обработчика'

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
exception isphere_exceptions.worker.InternalWorkerTimeout(*args, internal_code=510,  
**kwargs)
```

Базовые классы: `InternalWorkerErrorInterface`, `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE**= 'Превышен таймаут ответа обработчика'

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**log\_level**= 'error'

**livenessprobe**= *False*

```
exception isphere_exceptions.worker.InternalWorkerQueueFull(*args,  
internal_code=511, **kwargs)
```

Базовые классы: `InternalWorkerErrorInterface`, `ErrorNoReturnToQueue`

**DEFAULT\_MESSAGE**= 'Слишком много запросов в очереди источника'

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
log_level= 'error'
```

```
livenessprobe= False
```

```
exception isphere_exceptions.worker.InternalWorkerMaintenance(*args,  
internal_code=513, **kwargs)
```

Базовые классы: `InternalWorkerErrorInterface`, `ErrorNoReturnToQueue`

```
DEFAULT_MESSAGE= 'Сервис временно недоступен. Возможны технические работы'
```

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
log_level= 'warning'
```

```
livenessprobe= True
```

```
exception isphere_exceptions.worker.InternalWorkerOverload(*args,  
internal_code=514, **kwargs)
```

Базовые классы: `InternalWorkerErrorInterface`, `ErrorNoReturnToQueue`

```
DEFAULT_MESSAGE= 'Сервис перегружен запросами'
```

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
log_level= 'error'
```

```
livenessprobe= False
```

```
exception isphere_exceptions.worker.InternalWorkerNotPrepared(*args,  
internal_code=515, **kwargs)
```

Базовые классы: `InternalWorkerErrorInterface`, `ErrorNoReturnToQueue`

```
DEFAULT_MESSAGE= 'Произошла ошибка подготовки обработчика'
```

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
log_level= 'warning'
```

```
livenessprobe= False
```

```
exception isphere_exceptions.worker.UnknownError(*args, internal_code=599, **kwargs)
```

Базовые классы: `InternalWorkerErrorInterface`, `ErrorNoReturnToQueue`



**DEFAULT\_MESSAGE=** *'Неизвестная ошибка'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**log\_level=** *'error'*

**livenessprobe=** *False*

## Module contents

**class** `isphere_exceptions.CodeValidator`

Базовые классы: `object`

Дескриптор для валидации кода ошибки.

Ожидаемый диапазон кодов - [100, 1000)

**exception** `isphere_exceptions.ISphereException`(*message=None, code=500, internal\_code=None*)

Базовые классы: `Exception`

Базовый класс ошибки

**DEFAULT\_MESSAGE=** *'Событие i-sphere'*

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**EXCEPTION\_PREFIX=** *'INT'*

Дефолтный префикс для внутреннего кода ошибки

**TEMPLATE\_CONTEXT:** `Optional [ str ] = None`

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

**log\_level=** *'error'*

**livenessprobe=** *False*

**message**

Произвольный текст ошибки

**code=** *100*

Код ошибки. По умолчанию - 500. Значение, возвращаемое клиенту.

### **internal\_code**

Код внутренней ошибки. Не обязательно численно совпадает с `code`. Имеет в начале префикс `EXCEPTION_PREFIX`. Формат: `prefix-internal_code`

### **to\_response()**

*exception* **isphere\_exceptions.FailureEvent(\*args, \*\*kwargs)**

Базовые классы: `ISphereException`

Событие с отрицательным результатом.

Например, timeout подключения к БД, источник не отвечает и т.п.

**DEFAULT\_MESSAGE= 'Общая ошибка'**

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**EXCEPTION\_PREFIX= 'INT'**

Дефолтный префикс для внутреннего кода ошибки

**TEMPLATE\_CONTEXT:** `Optional [ str ] = None`

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

*exception* **isphere\_exceptions.SuccessEvent(\*args, code=200, \*\*kwargs)**

Базовые классы: `ISphereException`

Событие с положительным результатом.

Например, найдены данные или пользователь не существует в системе

**DEFAULT\_MESSAGE= 'Успешное событие'**

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**EXCEPTION\_PREFIX= 'SCS'**

Дефолтный префикс для внутреннего кода ошибки

**TEMPLATE\_CONTEXT:** `Optional [ str ] = None`

Дополнительные параметры, которые можно указать в сообщении ошибки при наличии шаблонизатора {} в `self.message`

```
log_level= 'info'
```

```
livenessprobe= True
```

```
exception isphere_exceptions.ErrorNoReturnToQueue(*args, code=200, **kwargs)
```

Базовые классы: `SuccessEvent`

Ошибка обработки данных при которой входные данные не нужно возвращать обратно в очередь

Все ошибки вида `SuccessEvent`

```
exception isphere_exceptions.ErrorReturnToQueue(*args, **kwargs)
```

Базовые классы: `FailureEvent`

Ошибка обработки данных при которой входные данные необходимо вернуть обратно в очередь

Все ошибки вида `FailureEvent`

```
exception isphere_exceptions.CommonException(*args, internal_code=500, **kwargs)
```

Базовые классы: `ErrorReturnToQueue`

Интерфейс общей ошибки при использовании зависимости (Keydb, Rabbitmq, Источник)

Код - 500.

```
DEFAULT_MESSAGE= 'Ошибка использования {}'
```

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
exception isphere_exceptions.ConfigurationInvalidException(*args, internal_code=501, **kwargs)
```

Базовые классы: `ErrorNoReturnToQueue`

Интерфейс ошибки конфигурирования зависимости (Keydb, Rabbitmq, Источник)

Код - 501

```
DEFAULT_MESSAGE  
= 'Заданная конфигурация подключения к {} некорректна. Проверьте введенные параметры и ENV'
```

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

```
livenessprobe= False
```

```
log_level= 'error'
```

**exception isphere\_exceptions.ConnectionException(\*args, internal\_code=502, \*\*kwargs)**

Базовые классы: `ErrorNoReturnToQueue`

Интерфейс ошибки подключения к зависимости (Keydb, Rabbitmq, Источник)

Код - 502

```
livenessprobe= False
```

```
log_level= 'error'
```

```
DEFAULT_MESSAGE= 'Ошибка подключения к {}'
```

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception isphere\_exceptions.TimeoutException(\*args, internal\_code=503, \*\*kwargs)**

Базовые классы: `ErrorReturnToQueue`

Интерфейс ошибки таймаута подключения к зависимости (Keydb, Rabbitmq, Источник)

Код - 503

```
DEFAULT_MESSAGE= 'Превышен таймаут подключения к {}, возможно недоступен'
```

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception isphere\_exceptions.OperationFailureException(\*args, internal\_code=504, \*\*kwargs)**

Базовые классы: `ErrorReturnToQueue`

Интерфейс ошибки выполнения действия у зависимости (Keydb, Rabbitmq, Источник)

Код - 504

```
DEFAULT_MESSAGE= 'При выполнении операции {} возникла ошибка'
```

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

**exception isphere\_exceptions.ParseErrorException(\*args, internal\_code=505, \*\*kwargs)**

Базовые классы: `ErrorReturnToQueue`

Интерфейс ошибки обработки данных от зависимости (Keydb, Rabbitmq, Источник)

Код - 505

**DEFAULT\_MESSAGE= 'Возникла ошибка во время обработки данных {}'**

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

*exception* **isphere\_exceptions.LockedException(\*args, internal\_code=506, \*\*kwargs)**

Базовые классы: `ErrorReturnToQueue`

Интерфейс ошибки временной блокировки от зависимости (Keydb, Rabbitmq, Источник)

Код - 506

**DEFAULT\_MESSAGE= 'Данная конфигурация {} временно заблокирована для источника'**

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе

*exception* **isphere\_exceptions.BlockedException(\*args, internal\_code=507, \*\*kwargs)**

Базовые классы: `ErrorReturnToQueue`

Интерфейс ошибки постоянной блокировки от зависимости (Keydb, Rabbitmq, Источник)

Код - 507

**DEFAULT\_MESSAGE= 'Данная конфигурация {} навсегда заблокирована для источника'**

Дефолтное сообщение об ошибке, в случае если не указан метод в конструкторе