

## Classes

```
class FastAGIProtocol(basic.LineOnlyReceiver)
class AMIProtocol(basic.LineOnlyReceiver)
class AMIFactory(protocol.ReconnectingClientFactory)
```

## Functions

### 1. class FastAGIProtocol(basic.LineOnlyReceiver)

```
→ def __init__(self, *args, **named)
→ def connectionMade(self)
→ def connectionLost(self, reason)
→ def onClose(self)
→ def lineReceived(self, line)
→ def sendCommand(self, commandString)
→ def checkFailure(self, result, failure='-1')
→ def resultAsInt(self, result)
→ def secondResultItem(self, result)
→ def resultPlusTimeoutFlag(self, resultLine)
→ def dateAsSeconds(self, date)
→ def onRecordingComplete(self, resultLine)
→ def onStreamingComplete(self, resultLine, skipMS=0)
→ def jumpOnError(self, reason, difference=100, forErrors=None)
→ def finish(self)
→ def answer(self)
→ def channelStatus(self, channel=None)
→ def onControlStreamFileComplete(self, resultLine)
→ def controlStreamFile(
    self, filename, escapeDigits, skipMS=0, ffChar='*',
    rewChar='#', pauseChar=None)
→ def databaseDel(self, family, key)
→ def databaseDeltree(self, family, keyTree=None)
→ def databaseGet(self, family, key)
→ def returnValue(resultLine)
→ def databaseSet(self, family, key, value)
→ def execute(self, application, *options, **kwargs)
→ def getData(self, filename, timeout=2.000, maxDigits=None)
→ def getOption(self, filename, escapeDigits, timeout=None)
→ def getVariable(self, variable)
→ def hangup(self, channel=None)
```

- def noop(self, message=None)
- def playback(self, filename, doAnswer=1)
- def receiveChar(self, timeout=None)
- def receiveText(self, timeout=None)
- def recordFile(
  - self, filename, format, escapeDigits, timeout=-1,
  - offsetSamples=None, beep=True, silence=None)
- def onResult(resultLine)
- def sayXXX(self, baseCommand, value, escapeDigits="")
- def sayAlpha(self, string, escapeDigits=None)
- def sayDate(self, date, escapeDigits=None)
- def sayDigits(self, number, escapeDigits=None)
- def sayNumber(self, number, escapeDigits=None)
- def sayPhonetic(self, string, escapeDigits=None)
- def sayTime(self, time, escapeDigits=None)
- def sayDateTime(self, time, escapeDigits=",", format=None, timezone=None)
- def sendImage(self, filename)
- def sendText(self, text)
- def setAutoHangup(self, time)
- def setCallerID(self, number)
- def setContext(self, context)
- def setExtension(self, extension)
- def setMusic(self, on=True, musicClass=None)
- def setPriority(self, priority)
- def setVariable(self, variable, value)
- def streamFile(self, filename, escapeDigits="", offset=0)
- def tddMode(self, on=True)
- def verbose(self, message, level=None)
- def waitForDigit(self, timeout)
- def wait(self, duration)

## 2. class InSequence(object)

- def \_\_init\_\_(self)
- def append(self, function, \*args, \*\*named)
- def \_\_call\_\_(self)
- def \_doSequence(self)
- def recordResult(self, result)
- def onActionSuccess(self, result, finalDF)
- def onActionFailure(self, reason, finalDF)

## 3. class FastAGIFactory(protocol.Factory)

- def \_\_init\_\_(self, mainFunction)