# Amazon Echo Javascript Skill Quick Ref Cheat Sheet

by PebbleSpot via cheatography.com/26328/cs/7494/

## Built in Slot Types

| | | |
|---|---|---|
| AMAZON.DATE | “today”, “tomorrow”, or “july” | “2015-07--00T9” |
| AMAZON.DURATION | “five minutes” | “PT5M” |
| AMAZON.FOUR_DIGIT_-NUMBER | "nineteen eighty four" | 1984 |
| AMAZON.NUMBER | “five” | "5" |
| AMAZON.US_CITY | all US cities with pop. > 100K | "Seattle" |
| AMAZON.US_STATE | US states, territories, and D.C. | "John" |
| AMAZON.US_FIRST-_NAME | recognizes popular first names | |

## Sample Utterances

```
GetHoroscope what is the horoscope for {Sign}
GetHor oscope what will the horoscope for {Sign} be on {Date}
GetHor oscope get me my horoscope
GetHor oscope {Sign}
...
GetLuc kyN umbers what are my lucky numbers
GetLuc kyN umbers tell me my lucky numbers
...
MatchSign do {First Sign} and {Secon dSign} get along
MatchSign what is the relati onship between {First Sign} and {Secon dSign}
```

The name of the intent on the left.
The phrase a user might speak to signal that intent on the right.

Slots {SlotName} are arguments to intents.

## Custom Slots Example

```
{ "intents": [
       { " int ent ": " Get Hor osc ope ",
            " slo ts": [
               { " nam e": " Sig n",
                    " typ e": " LIS T_O F_S IGN S"
},
```

## Custom Slots Example (cont)

```
>        { "name": "Date",
          "type": "AMAZON.DATE" }
       ] },
    { "intent": "MatchSign",
      "slots": [
       { "name": "FirstSign",
         "type": "LIST_OF_SIGNS" },
       { "name": "SecondSign",
         "type": "LIST_OF_SIGNS" }
       ] },
    { "intent": "GetLuckyNumbers" }
] }
```

https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/-docs/defining-the-voice-interface

## Built-In Intents

| | |
|---|---|
| AMAZON.CancelIntent | "Cancel," "Never mind," "Forget it" |
| AMAZON.HelpIntent | "Help," "Help me," "Can you help me" |
| AMAZON.NoIntent | "No," "No thanks |
| AMAZON.RepeatIntent | "Repeat," "Say that again," "Repeat that" |
| AMAZON.StartOverIntent | "Start over," "Restart," "Start again" |
| AMAZON.StopIntent | "Stop," "Off," "Shut up" |
| AMAZON.YesIntent | "Yes," "Yes please," "Sure" |

https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/-docs/implementing-the-built-in-intents

## Implement Built-in Intent Schema

```
{
     " int ent s": [
        { " int ent ": " Get Fir stE ven tIn ten t",
               " slo ts": [
                  { " nam e": " day ",
```



By **PebbleSpot**

cheatography.com/pebblespot/

Not published yet.
Last updated 11th May, 2016.


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

## Implement Built-in Intent Schema (cont)

```
>          "type": "AMAZON.DATE" }
     ]
   },
   { "intent": "GetNextEventIntent" },
   { "intent": "AMAZON.HelpIntent" },
   { "intent": "AMAZON.StopIntent" }
 ]
}
```

## Implement Built-in Intent Code

```
@Override
public Speech let Res ponse onInte nt( final
Intent Request request, final Session session)
                throws Speech let Exc eption {
        log.in fo( " onI ntent reques tId={},
sessio nId ={} ", reques t.g etR equ est Id(),
                        ses sio n.g etS ess ion -
Id());

        Intent intent = reques t.g etI nte nt();
        String intentName = intent.ge tNa me();

       if ("Ge tFi rst Eve ntI nte nt".e qu als -
(in ten tName)) {
                return handle Fir stE ven tRe que -
st( intent, session);
       } else if ("Ge tNe xtE ven tIn ten t".e -
qua ls( int ent Name)) {
                return handle Nex tEv ent Req ues -
t(s ess ion);
       } else if ("AM AZO N.H elp Int ent ".eq -
ual s(i nte ntN ame)) {
               // Handling for the help intent goes
here
       } else if ("AM AZO N.S top Int ent ".eq -
ual s(i nte ntN ame)) {
               // Handling for the stop intent goes
here.
       } else {
                throw new Speech let Exc ept ion -
("In valid Intent ");
       }
}
```

## Response Object

```
{
    ver sion: '1.0',
    res ponse: {
        out put Speech: { // Always part of the
response
            type: 'Plain Text', // Always this
value
            text: 'Example speech' // Text that
will be spoken back to user
       },
        sho uld End Ses sion: true, // Always part
of the respon se- -bo olean for if session has
ended (no more user input)
        rep rompt: { // Optional, if you want to
customize what Alexa says if user doesn't respond
correc tly/at all
            out put Speech: {
                type: 'Plain Text',
                text: 'Example reprompt'
           }
       },
        card: { // Optional, if you want to show a
card in the Echo app
            type: 'Simple',
            title: 'Example title',
            con tent: 'Example card content'
       }
   },
    ses sio nAt tri butes: {} // If you need some
attributes to be kept in the session
}
```



Sponsored by **Readable.com**
Measure your website readability!
https://readable.com