

Programming Conventions

This manual contains the **Naming Conventions** when creating your solutions, projects, datastore classes, collections, datastore class methods, datasources, and attributes.

The **Reserved Keywords** section informs you of all the words that are reserved and not used in your variable names, functions, datastore classes, and attributes.

Naming Conventions

Since Wakanda solutions, projects, datastore classes, collections, datastore class methods, datasources, and attributes are handled through JavaScript, they must comply with JavaScript's naming rules: names and IDs must begin with a letter (A-Z, a-z) and can contain letters (A-Z, a-z), digits (0-9), and underscores. Accented and non-Roman characters are accepted; however, the following characters are not: !@#\$%^&*()+= -[]';,./{}|\":<>?~.

Widgets must follow the standard HTML specifications: widget IDs must begin with a letter (A-Z, a-z) and can contain letters (A-Z, a-z), digits (0-9), and underscores. Characters, such as !@#\$%^&*()+= -[]';,./{}|\":<>?~, as well as accented and non-Roman characters are not allowed for the ID of a widget.

Spaces in any of the names or IDs are also not allowed. If you include one, the Datastore Model Designer and GUI Designer will automatically replace it with either an underscore or a hyphen.

Following JavaScript's conventions, variables and functions begin with a lowercase letter while constructors and class names begin with an uppercase letter and the rest in camel case (i.e., all the spaces in the words are removed and each word is capitalized). In Wakanda, we have defined a few of our own naming conventions along these same lines:

- **Datastore Class:** The first letter must be capitalized to follow the logic of the datasource whose first letter must be lowercase. For the "LinItem" datastore class, the datasource is named "linItem," by default. Generally, the class name is singular.
- **Collection:** Collection names follow the same rule as datastore class names. If you have a datastore class named "LinItem," Wakanda generates "LinItemCollection" as its collection name. You can also rename it to "LinItems" (the plural form of the datastore class name), if you prefer.
- **Attribute:** It is suggested that the first letter of an attribute be lowercase so that you can easily distinguish datastore classes/collections from attributes. For example, "firstName." Although allowed, avoid naming attributes the same as datastore classes
- **Datastore Class Method:** It is suggested that the name of a datastore class method be written in camel case and can begin with either a lowercase or an uppercase letter. For example, "checkEmployee" or "CheckEmployee."
- **Widget:** It is suggested that the first letter of the widget ID be lowercase and that all other words are in camel case. For example, "myGrid."
- **Datasource:** The first letter of a datasource name must be lowercase to follow the logic of the datastore class whose first letter must be in uppercase. For the "LinItem" datastore class, the datasource is named "linItem," by default.

In addition, you must avoid [Reserved Keywords](#) when naming objects.

Reserved Keywords

This section lists the keywords that must not be used as names for **datastore classes**, **attributes** or **methods**. It also provides an extensive list of JavaScript, ECMAScript and Wakanda keywords that should not be used as **variable names**.

*Warning: If you have checked the **Allow Global Access** option in the **Datastore Model Designer** for your model or a datastore class, pay attention to the fact that it will be set at the global object level of the application. In this configuration, you must not use any JavaScript and ECMAScript reserved keywords (see the list below) in order to avoid name conflicts. For more information on the "Allow Global Access" option, please refer to the [Datastore Class Properties](#).*

Forbidden Keywords

The following table lists keywords that cannot be used as attribute names, method names or datastore class names:

Keywords	Attributes	Datastore class methods	Datastore classes
add	X		
addEntity	X		
addListener	X		
addNewElement	X		
all	X	X	
allEntities	X		
and	X		
autoDispatch	X		
average	X	X	
buildFromSelection	X		
calculated	X		
callMethod	X		
close			X
compute	X	X	
count	X	X	
createEntity	X	X	
createEntityCollection	X	X	
dataClasses			X
declareDependencies	X		
dispatch	X		
distinctValues	X	X	
flushCache			X
filterQuery	X		
find	X	X	
first	X	X	
forEach	X	X	
fromArray	X	X	
getAttribute	X		
getAttributeNames	X		
getAttributeValue	X		
getClassAttributeByName	X		
getClassTitle	X		
getCurrentElement	X		
getDataClass	X		

getDataFolder			X
getElement	X		
getEntityCollection	X		
getFragmentation	X	X	
getID	X		
getKey	X		
getModelFolder			X
getName	X	X	X
getOldAttributeValue	X		
getPosition	X		
getScope	X	X	
getSelection	X		
getStamp	X		
getTempFolder			X
getTimeStamp	X		
isLoading	X		
isModified	X		
isNew	X		
isNewElement	X		
length	X		
max	X	X	
min	X	X	
minus	X		
mustResolveOnFirstLevel	X		
newEntity	X		
next	X		
noEntities	X		
or	X		
orderBy	X	X	
parentCategory	X		
query	X	X	
queryPath	X		
queryPlan	X		
refresh	X		
release	X		
remove	X	X	
removeAllListeners	X		
removeCurrent	X		
removeListener	X		
resolveSource	X		
save	X		
select	X		
selectByKey	X		
selectNext	X		
selectPrevious	X		
serverRefresh	X		
setAutoSequenceNumber	X	X	

setCurrentEntity	X	
setDisplayLimits	X	
setEntityCollection	X	
sum	X	X
sync	X	
toArray	X	X
toString	X	X
valid	X	
validate	X	

Note: This list is not final and may evolve along with Wakanda Server-side APIs.

Reserved Keywords

The following keywords must not be used as variable or function names. These keywords are reserved by JavaScript, ECMAScript (versions 3, 5, and 5.1) or by Wakanda itself (as specified in the Reserved by column).

Keyword	Reserved by
abstract	ECMAScript
administrator	Wakanda
alert	JavaScript
Anchor	JavaScript
application	Wakanda
Area	JavaScript
arguments	JavaScript
Array	JavaScript
assign	JavaScript
blob	JavaScript
blur	JavaScript
bool	JavaScript
boolean	ECMAScript
Boolean	JavaScript
break	ECMAScript
Button	JavaScript
byte	ECMAScript/JavaScript
callee	JavaScript
caller	JavaScript
captureEvents	JavaScript
case	ECMAScript
catch	ECMAScript
char	ECMAScript
Checkbox	JavaScript
class	ECMAScript
clearInterval	JavaScript
clearTimeout	JavaScript
close	JavaScript
closed	JavaScript
confirm	JavaScript
console	Wakanda

const	ECMAScript
constructor	JavaScript
continue	ECMAScript
dataService	Wakanda
date/Date	JavaScript
debugger	ECMAScript
default	ECMAScript
defaultStatus	JavaScript
delete	ECMAScript
directory	Wakanda
do	ECMAScript
document/Document	JavaScript
double	ECMAScript
ds	Wakanda
duration	JavaScript
Element	JavaScript
else	ECMAScript
enum	ECMAScript
escape	JavaScript
eval	JavaScript
export	ECMAScript
exports	JavaScript
extends	ECMAScript
FALSE	JavaScript
FileUpload	JavaScript
final	ECMAScript
finally	ECMAScript
find	JavaScript
float	ECMAScript
focus	JavaScript
for	ECMAScript
Form	JavaScript
Frame	JavaScript
frames	JavaScript
function	ECMAScript
Function	JavaScript
getClass	JavaScript
goto	ECMAScript
Hidden	JavaScript
history/History	JavaScript
home	JavaScript
httpServer	Wakanda
if	ECMAScript
image/Image	JavaScript
implements	ECMAScript
import	ECMAScript
in	ECMAScript

Infinity	JavaScript
innerHeight	JavaScript
innerWidth	JavaScript
instanceof	ECMAScript
int	ECMAScript
interface	ECMAScript
isFinite	JavaScript
isNaN	JavaScript
java	JavaScript
JavaArray	JavaScript
JavaClass	JavaScript
JavaObject	JavaScript
JavaPackage	JavaScript
length	JavaScript
Link	JavaScript
location/Location	JavaScript
locationbar	JavaScript
long	ECMAScript/JavaScript
long64	JavaScript
Math	JavaScript
menubar	JavaScript
MimeType	JavaScript
moveBy	JavaScript
moveTo	JavaScript
NaN	JavaScript
navigate	JavaScript
navigator/Navigator	JavaScript
native	ECMAScript
netscape	JavaScript
new	ECMAScript
null	JavaScript
number/Number	JavaScript
Object	JavaScript
onBlur	JavaScript
onError	JavaScript
onFocus	JavaScript
onLoad	JavaScript
onUnload	JavaScript
open	JavaScript
opener	JavaScript
Option	JavaScript
os	Wakanda
outerHeight	JavaScript
outerWidth	JavaScript
package	ECMAScript
Packages	JavaScript
pageXoffset	JavaScript

pageYoffset	JavaScript
parent	JavaScript
parseFloat	JavaScript
parseInt	JavaScript
Password	JavaScript
personalbar	JavaScript
Plugin	JavaScript
private	ECMAScript
print	JavaScript
process	Wakanda
prompt	JavaScript
protected	ECMAScript
prototype	JavaScript
public	ECMAScript
Radio	JavaScript
ref	JavaScript
RegExp	JavaScript
releaseEvents	JavaScript
Reset	JavaScript
resizeBy	JavaScript
resizeTo	JavaScript
return	ECMAScript
routeEvent	JavaScript
rpcService	Wakanda
scroll	JavaScript
scrollbars	JavaScript
scrollBy	JavaScript
scrollTo	JavaScript
Select	JavaScript
self	JavaScript
sessionStorage	Wakanda
setInterval	JavaScript
setTimeout	JavaScript
settings	Wakanda
short	ECMAScript
solution	Wakanda
static	ECMAScript
status	JavaScript
statusbar	JavaScript
stop	JavaScript
storage	Wakanda
String	JavaScript
Submit	JavaScript
sun	JavaScript
super	ECMAScript
synchronized	ECMAScript
switch	ECMAScript

taint	JavaScript
Text	JavaScript
Textarea	JavaScript
this	ECMAScript
throw	ECMAScript
throws	ECMAScript
toolbar	JavaScript
top	JavaScript
toString	JavaScript
transient	ECMAScript
TRUE	JavaScript
try	ECMAScript
typeof	ECMAScript
unescape	JavaScript
untaint	JavaScript
unwatch	JavaScript
uuid	JavaScript
valueOf	JavaScript
var	ECMAScript
void	ECMAScript
volatile	ECMAScript
watch	JavaScript
webAppService	Wakanda
while	ECMAScript
wildchar	Wakanda
window/Window	JavaScript
with	ECMAScript
word	JavaScript