



SearchNode Tool

SearchNode **fuzzy-matches** entity queries to relevant RPG nodes.



Tool Call

```
SearchNode(  
  feature_terms="expression serialization"  
)
```

Search Result:

Found method

```
`django/db/models/expressions.py:Base  
Expression.__getstate__`.
```

It Functionality Features:

DatabaseORMAndMigrationSystem/sql
utilities/expressions/prepare state for
serialization

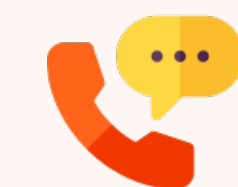
```

```
166 | def __getstate__(self):
167 | state = self.__dict__.copy()
168 | state.pop('convert_value',
None)
169 | return state
```
```



FetchNode Tool

FetchNode retrieves **precise metadata** for specified RPG node.



Tool Call

```
FetchNode(code_entities=  
"sympy/physics/vector/vector.py:_  
check_vector")
```

Search Result

Found function

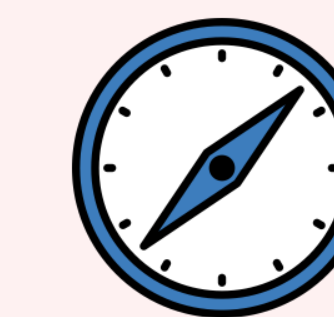
```
`sympy/physics/vector/vector.py:_check  
_vector`.
```

It Functionality Features:

SymbolicComputation/vector
algebra/vector operations/validate input
is a Vector instance

```

```
723 | def _check_vector(other):
724 | if not isinstance(other, Vector):
725 | raise TypeError('A Vector must
be supplied')
726 | return other
```
```



ExploreRPG

ExploreRPG **traverses the RPG** to reveal edge-linked relations between connected node.



Tool Call

```
ExploreRPG(start_code_entities=  
["sympy/core/sympify.py:kernS"]  
)
```

sympy/core/sympify.py:kernS [F1]

|- invokes --- sympy/core/
sympify.py:sympify [F2]
|- invokes --sympy/parsing/sy
mpy_parser.py:parse_expr [F3]

Feature Paths (Legend):

SymbolicComputation

├── algebraic constructs
│ ├── sympification operations
│ │ └── sympify
│ └── extend conversion with
custom objects [F2]
└── prevent autosimplification
from merging operations [F1]