```
{
  <AreaChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickArea": PropTypes.func,
      "onClickLine": PropTypes.func,
      "onHoverArea": PropTypes.func,
      "onHoverLine": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <BarChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickBar": PropTypes.func,
      "onHoverBar": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <BillboardChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickBillboard": PropTypes.func,
      "onHoverBillboard": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <BlockText>: {
    "defaultProps": "undefined",
    "propTypes": {
      "children": PropTypes.element,
      "style": PropTypes.object
    }
  },
  <Button>: {
    "SIZE_TYPE": {
      "LARGE": "large",
      "NORMAL": "normal",
      "SLIM": "slim"
    },
    "TAG_TYPE": {
      "A": "a",
      "BUTTON": "button"
    },
    "VARIANT_TYPE": {
      "DEFAULT": "default",
      "DESTRUCTIVE": "destructive",
      "PLAIN": "plain",
      "PLAIN_NEUTRAL": "plainNeutral",
      "PRIMARY": "primary"
    },
    "defaultProps": {
      "sizeType": "normal",
      "tagType": "button",
      "variantType": "default"
    },
    "propTypes": {
      "children": PropTypes.element,
      "className": PropTypes.string,
      "disabled": PropTypes.bool,
      "icon": PropTypes.oneOf(['dataviz_dataviz_bar-
chart','dataviz_dataviz_chart','dataviz_dataviz_chart_a-a…),
      "loading": PropTypes.bool,
      "onClick": PropTypes.func,
      "sizeType": PropTypes.oneOf(['slim','normal','large']),
      "style": PropTypes.object,
      "tagType": PropTypes.oneOf(['button','a']),
      "to": PropTypes.string,
      "variantType":
PropTypes.oneOf(['default','primary','plain','plainNeutral','destructive'])
    }
  },
  <ChartGroup>: {
    "defaultProps": "undefined",
    "propTypes": "undefined"
  },
  <Checkbox>: {
    "defaultProps": "undefined",
    "propTypes": {
      "checked": PropTypes.bool,
      "className": PropTypes.string,
      "defaultChecked": PropTypes.bool,
      "disabled": PropTypes.bool,
      "id": PropTypes.string,
      "indeterminate": () => { ... },
      "label": PropTypes.string,
      "name": PropTypes.string,
      "onChange": PropTypes.func,
      "style": PropTypes.object
    }
  },
  <Dialog>: {
    "defaultProps": {
      "hidden": false,
      "onHide": () => { ... },
      "onHideEnd": () => { ... },
      "onShow": () => { ... },
      "onShowEnd": () => { ... }
    },
    "propTypes": {
      "hidden": PropTypes.bool,
      "onClose": PropTypes.func.isRequired,
      "onHide": PropTypes.func,
      "onHideEnd": PropTypes.func,
      "onShow": PropTypes.func,
      "onShowEnd": PropTypes.func
    }
  },
  <DisplayText>: {
    "defaultProps": "undefined",
    "propTypes": {
      "children": PropTypes.element,
      "style": PropTypes.object
    }
  },
  <EntitiesByDomainTypeQuery>: {
    "defaultProps": {
      "includeTags": false
    },
    "propTypes": {
      "customAttributesFragment": PropTypes.object,
      "entityDomain": PropTypes.string.isRequired,
      "entityType": PropTypes.string.isRequired,
      "filters": PropTypes.arrayOf(),
      "includeTags": PropTypes.bool
    }
  },
  <EntitiesByIdsQuery>: {
    "defaultProps": {
      "includeRelationships": false,
      "includeTags": true
    },
    "propTypes": {
      "customAttributesFragment": PropTypes.object,
      "customAttributesOutlineFragment": PropTypes.object,
      "entityIds": PropTypes.arrayOf().isRequired,
      "includeRelationships": PropTypes.bool,
      "includeTags": PropTypes.bool
    }
  },
  <EntitiesByNameQuery>: {
    "defaultProps": {
      "includeTags": false
    },
    "propTypes": {
      "customAttributesFragment": PropTypes.object,
      "filters": PropTypes.arrayOf(),
      "includeTags": PropTypes.bool,
      "name": PropTypes.string.isRequired
    }
  },
  <EntityByIdQuery>: {
    "defaultProps": {
      "includeRelationships": false,
      "includeTags": true
    },
    "propTypes": {
      "customAttributesFragment": PropTypes.object,
      "customAttributesOutlineFragment": PropTypes.object,
      "entityId": PropTypes.string.isRequired,
      "includeRelationships": PropTypes.bool,
      "includeTags": PropTypes.bool
    }
  },
  <EntityCountQuery>: {
    "defaultProps": "undefined",
    "propTypes": {
      "filters": PropTypes.arrayOf()
    }
  },
  <EntitySearchQuery>: {
    "defaultProps": {
      "includeCount": false,
      "includeResults": true,
      "includeTags": false
    },
    "propTypes": {
      "customAttributesFragment": PropTypes.object,
      "entityDomain": PropTypes.string,
      "entityType": PropTypes.string,
      "filters": PropTypes.arrayOf(),
      "includeResults": PropTypes.bool,
      "includeTags": PropTypes.bool,
      "name": PropTypes.string
    }
  },
  <FunnelChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickFunnel": PropTypes.func,
      "onHoverFunnel": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <Grid>: {
    "defaultProps": "undefined",
    "propTypes": {
      "children": () => { ... },
      "className": PropTypes.string,
      "style": PropTypes.object
    }
  },
  <GridItem>: {
    "defaultProps": "undefined",
    "propTypes": {
      "children": PropTypes.any,
      "className": PropTypes.string,
      "collapseGapAfter": PropTypes.bool,
      "collapseGapBefore": PropTypes.bool,
      "columnEnd": (props, propName, componentName) => { ... },
      "columnSpan": (props, propName, componentName) => { ... },
      "columnStart": (props, propName, componentName) => { ... },
      "style": PropTypes.object
    }
  },
  <HeatmapChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickHeatmap": PropTypes.func,
      "onHoverHeatmap": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <HistogramChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickHistogram": PropTypes.func,
      "onHoverHistogram": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <Icon>: {
    "SIZE_TYPE": {
      "LARGE": 32,
      "NORMAL": 16,
      "SMALL": 8
    },
    "defaultProps": {
      "color": "inherit",
      "name": null,
      "sizeType": 16
    },
    "propTypes": {
      "className": PropTypes.string,
      "color": PropTypes.string,
      "name": PropTypes.oneOf(['dataviz_dataviz_bar-
chart','dataviz_dataviz_chart','dataviz_dataviz_chart_a-a…).isRequired,
      "sizeType": PropTypes.oneOf([8,16,32]),
      "style": PropTypes.object
    }
  },
  <JsonChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "query": PropTypes.string
    }
  },
  <LineChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickArea": PropTypes.func,
      "onClickLine": PropTypes.func,
      "onHoverArea": PropTypes.func,
      "onHoverLine": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <Link>: {
    "defaultProps": "undefined",
    "propTypes": "undefined"
  },
  <NerdGraphQuery>: {
    "defaultProps": "undefined",
    "propTypes": {
      "query": PropTypes.object,
      "variables": PropTypes.object
    }
  },
  <NerdStoreAccountCollectionMutation>: {
    "ACTION_TYPE": {
      "DELETE_COLLECTION": "DELETE_COLLECTION",
      "DELETE_DOCUMENT": "DELETE_DOCUMENT",
      "WRITE_DOCUMENT": "WRITE_DOCUMENT"
    },
    "defaultProps": "undefined",
    "propTypes": {
      "action":
PropTypes.oneOf(['DELETE_COLLECTION','DELETE_DOCUMENT','WRITE_DOCUMENT']).i
sRequired,
      "collection": PropTypes.string,
      "document": () => { ... },
      "documentId": PropTypes.string
    }
  },
  <NerdStoreAccountCollectionQuery>: {
    "defaultProps": "undefined",
    "propTypes": "undefined"
  },
  <NerdStoreEntityCollectionMutation>: {
    "ACTION_TYPE": {
      "DELETE_COLLECTION": "DELETE_COLLECTION",
      "DELETE_DOCUMENT": "DELETE_DOCUMENT",
      "WRITE_DOCUMENT": "WRITE_DOCUMENT"
    },
    "defaultProps": "undefined",
    "propTypes": {
      "action":
PropTypes.oneOf(['DELETE_COLLECTION','DELETE_DOCUMENT','WRITE_DOCUMENT']).i
sRequired,
      "collection": PropTypes.string,
      "document": () => { ... },
      "documentId": PropTypes.string
    }
  },
  <NerdStoreEntityCollectionQuery>: {
    "defaultProps": "undefined",
    "propTypes": "undefined"
  },
  <NerdStoreUserCollectionMutation>: {
    "ACTION_TYPE": {
      "DELETE_COLLECTION": "DELETE_COLLECTION",
      "DELETE_DOCUMENT": "DELETE_DOCUMENT",
      "WRITE_DOCUMENT": "WRITE_DOCUMENT"
    },
    "defaultProps": "undefined",
    "propTypes": {
      "action":
PropTypes.oneOf(['DELETE_COLLECTION','DELETE_DOCUMENT','WRITE_DOCUMENT']).i
sRequired,
      "collection": PropTypes.string,
      "document": () => { ... },
      "documentId": PropTypes.string
    }
  },
  <NerdStoreUserCollectionQuery>: {
    "defaultProps": "undefined",
    "propTypes": "undefined"
  },
  <NrqlQuery>: {
    "FORMAT_TYPE": {
      "CHART": "chart",
      "RAW": "raw"
    },
    "defaultProps": {
      "formatType": "chart"
    },
    "propTypes": {
      "accountId": PropTypes.number.isRequired,
      "children": PropTypes.func.isRequired,
      "formatType": PropTypes.oneOf(['chart','raw']),
      "query": PropTypes.string.isRequired
    }
  },
  <PieChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickPie": PropTypes.func,
      "onHoverPie": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <ProgressChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickProgress": PropTypes.func,
      "onHoverProgress": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <Radio>: {
    "defaultProps": "undefined",
    "propTypes": {
      "checked": PropTypes.bool,
      "className": PropTypes.string,
      "disabled": PropTypes.bool,
      "label": PropTypes.string,
      "name": PropTypes.string,
      "onChange": PropTypes.func,
      "style": PropTypes.object,
      "value": PropTypes.any
    }
  },
  <ScatterChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickScatter": PropTypes.func,
      "onHoverScatter": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <SparklineChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickSparkline": PropTypes.func,
      "onHoverSparkline": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <Spinner>: {
    "TYPE_CIRCLE": "circle",
    "TYPE_INLINE": "inline",
    "defaultProps": {
      "type": "circle"
    },
    "propTypes": {
      "className": PropTypes.string,
      "style": PropTypes.object,
      "type": PropTypes.oneOf(['circle','inline']).isRequired
    }
  },
  <Stack>: {
    "ALIGNMENT_TYPE": {
      "BASELINE": "baseline",
      "CENTER": "center",
      "FILL": "fill",
      "LEADING": "leading",
      "TRAILING": "trailing"
    },
    "DIRECTION_TYPE": {
      "HORIZONTAL": "horizontal",
      "VERTICAL": "vertical"
    },
    "DISTRIBUTION_TYPE": {
      "CENTER": "center",
      "FILL": "fill",
      "FILL_EVENLY": "fillEvenly",
      "LEADING": "leading",
      "TRAILING": "trailing"
    },
    "SPACING_TYPE": {
      "EXTRA_LOOSE": "extraLoose",
      "LOOSE": "loose",
      "NONE": "none",
      "NORMAL": "normal",
      "TIGHT": "tight"
    },
    "defaultProps": {
      "alignmentType": "leading",
      "directionType": "horizontal",
      "distributionType": "leading",
      "spacingType": "normal",
      "wrap": false
    },
    "propTypes": {
      "alignment":
PropTypes.oneOf(['leading','trailing','center','fill','baseline']),
      "children": () => { ... },
      "className": PropTypes.string,
      "direction": PropTypes.oneOf(['horizontal','vertical']),
      "distributionType":
PropTypes.oneOf(['leading','trailing','center','fill','fillEvenly']),
      "spacingType":
PropTypes.oneOf(['none','tight','normal','loose','extraLoose']),
      "style": PropTypes.object,
      "wrap": PropTypes.bool
    }
  },
  <StackItem>: {
    "defaultProps": {
      "grow": false,
      "shrink": false
    },
    "propTypes": {
      "className": PropTypes.string,
      "grow": PropTypes.bool,
      "shrink": PropTypes.bool
    }
  },
  <StackedBarChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickStackedBar": PropTypes.func,
      "onHoverStackedBar": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <TableChart>: {
    "defaultProps": "undefined",
    "propTypes": {
      "accountId": PropTypes.number,
      "data": PropTypes.any,
      "onClickTable": PropTypes.func,
      "onHoverTable": PropTypes.func,
      "query": PropTypes.string
    }
  },
  <Tabs>: {
    "defaultProps": "undefined",
    "propTypes": {
      "activeTabId": PropTypes.any,
      "children": () => { ... },
      "className": PropTypes.string,
      "defaultActiveTabId": PropTypes.any,
      "onChange": PropTypes.func,
      "style": PropTypes.object
    }
  },
  <TabsItem>: {
    "defaultProps": "undefined",
    "propTypes": {
      "children": PropTypes.any,
      "className": PropTypes.string,
      "disabled": PropTypes.bool,
      "id": PropTypes.any.isRequired,
      "label": PropTypes.string.isRequired,
      "style": PropTypes.object,
      "tabIndex": PropTypes.number
    }
  },
  <TextField>: {
    "TYPE": {
      "EMAIL": "email",
      "PASSWORD": "password",
      "SEARCH": "search",
      "TEXT": "text",
      "URL": "url"
    },
    "defaultProps": {
      "type": "text"
    },
    "propTypes": {
      "autoFocus": PropTypes.bool,
      "className": PropTypes.string,
      "defaultValue": PropTypes.any,
      "disabled": PropTypes.bool,
      "id": PropTypes.string,
      "label": PropTypes.string,
      "loading": PropTypes.bool,
      "multiline": PropTypes.bool,
      "name": PropTypes.string,
      "onBlur": PropTypes.func,
      "onChange": PropTypes.func,
      "onFocus": PropTypes.func,
      "placeholder": PropTypes.string,
      "readOnly": PropTypes.bool,
      "style": PropTypes.object,
      "type": PropTypes.oneOf(['email','password','search','text','url']),
      "value": PropTypes.string
    }
  },
  <Toast>: {
    "defaultProps": {
      "actions": {},
      "autoDismiss": true,
      "description": "",
      "type": "normal"
    },
    "propTypes": {
      "actions": PropTypes.arrayOf(),
      "autoDismiss": PropTypes.bool,
      "description": PropTypes.string,
      "onDestroy": PropTypes.func.isRequired,
      "title": PropTypes.string.isRequired,
      "type": PropTypes.oneOf(['normal','critical'])
    }
  },
  <Tooltip>: {
    "PLACEMENT": {
      "BOTTOM": "bottom",
      "LEFT": "left",
      "RIGHT": "right",
      "TOP": "top"
    },
    "defaultProps": {
      "placement": "top"
    },
    "propTypes": {
      "children": PropTypes.element,
      "className": PropTypes.string,
      "placement": PropTypes.oneOf(['top','right','bottom','left']),
      "text": PropTypes.string
    }
  },
  launcher: {
    "setUrlState": (state, urlStateOptions) => { ... }
  },
  navigation: {
    "getOpenCardLocation": (cards) => { ... },
    "getOpenEntityLocation": (entity) => { ... },
    "getOpenLauncherLocation": (launcher) => { ... },
    "getOpenNerdletLocation": (nerdlet) => { ... },
    "getOpenOverlayLocation": (overlay) => { ... },
    "getReplaceNerdletLocation": (nerdlet) => { ... },
    "openCard": (cards) => { ... },
    "openEntity": () => { ... },
    "openLauncher": (launcher) => { ... },
    "openNerdlet": (nerdlet) => { ... },
    "openOverlay": (overlay) => { ... },
    "replaceNerdlet": (nerdlet) => { ... }
  },
  nerdlet: {
    "setUrlState": (state, urlStateOptions) => { ... }
  }
}
```